A large, stylized letter 'A' in black, with a thick black outline and a solid black fill. It is positioned on the left side of the page, partially overlapping the teal background.

August 1988

Vol. 1 N° 11

Price £1.50

Archive

The Subscription Magazine for Archimedes Users

A large, teal-colored jagged shape on the left side of the page, resembling a stylized 'E' or a series of connected triangles. It is positioned vertically, spanning most of the page height.

What's on the Menu? – WIMPs!

Floating Point Assembler

Screen Compaction Program

Logo, Presenter, Assembly Book, Utilities, ArcTerm
& U-Connect Reviews

Inside the MEMC Memory Controller

Oramics – Graphics to Sound

Using Hard Discs

More on First Word Plus, View, SigmaSheet &
GraphicWriter

Archive prices up again?

I'm afraid that we're having to think in terms of putting up the price of Archive again – we're only just making ends meet because the number of subscriptions hasn't risen quite as fast as it seemed it was going to do. So, from the 1st September, a year's subscription (12 issues) will go up to £14.50 with corresponding rises in the overseas rates.

Is the print size getting smaller?

If you are having trouble reading some of the articles, it's because we've reduced the point size yet again to try to fit in more material. It means that you get better value for money, but you may feel that it is getting too difficult to read. If you think it is too small, let me know. The following paragraph is in the smaller size we've used on some of the articles.

If you are having trouble reading some of the articles, it's because we've reduced the point size yet again to try to fit in more material. It means that you get better value for money, but you may feel that it is getting too difficult to read. If this is too small, let me know.

Where is Norwich Computer Services?

We're not difficult to find really – Mile End Road is the section of the Norwich Ring Road going west from where the A11 (Newmarket Road) joins it. Don't look for a shop or an office block though – Number 18 is actually a private house. We used to have a small terraced house and rented an office in the centre of Norwich but we decided to put the money that went into rent on the office into a mortgage for a house that was big enough for both. Archive operates from a huge room looking out onto the garden, so as I write I can watch the wild life – and the traffic!

Do pop in and see us if you're passing through – if you drop hints like “we've just driven all the way up from London”, you might even get offered a cup of coffee, especially if you buy some software!

Because we operate from a single room, it always amuses me when people ring up and ask to be put through to the Accounts Department or the Despatch Department – there's still just Sue and myself running the business. Mind you, we do have a lot of voluntary help from people like Adrian Look, Carl Wright (who runs Eureka!) and my in-laws – we can't thank them enough for all they do. And, as always, many thanks to the Managing Director, too!

Archive

Volume 1 • Nº 11 • August 1988

Contents

Hardware & Software Available ...2	Floating Point Assembler28
Comment Column3	Unpacking Packed BCD32
Help!!4	Alphabetic Alternatives.....34
Matters Arising4	Compacting Screens36
Hints & Tips5	Assembly Language Book39
MS-DOS Column8	Assembly Programming – 240
BASIC V Utilities Forum12	Communications Software.....42
Contact Box12	First Word Plus Forum45
Languages Corner – Pascal14	Presentation Graphics46
Hardware Column16	Programmers' Utilities.....48
Small Ad's18	Logotron Logo Review50
Using Hard Discs20	GraphicWriter/Wordwise Conv...51
What's on the Menu? – WIMPs..24	SigmaSheet Comments52
Oramics – Graphics to Sound27	Fact-File53

Hardware & Software Available

• **GammaPlot released** – Minerva Systems have now released their GammaPlot package at £69.95. (£64 through Archive.) It supports 6 main graph and chart types and has extensive art facilities which can be used to enhance the charts or can be used on their own as an art package to rival many of the others currently available. (Review in next month's magazine, hopefully.)

• **System Delta Plus Reference Manual** now also available from Minerva at £29.95 (£27 through Archive.) giving advanced programming techniques for this powerful database system.

• **3-D Spreadsheet** (Now there's a concept to grapple with!) Matrix 3, £95 + VAT from Cambridge Microsystems is a three-dimensional scientific spreadsheet, 10,000 rows by 10,000 columns by 100 pages – cells can even contain programs as well as straightforward arithmetic, trigonometric, hyperbolic, logical and statistical functions.

• **Archimedes Assembly Language** by Mike Ginns – A Dabhand Guide to assembly language, specifically written for the Archimedes costing £14.95 (£13.50 through Archive.). See the review on page 39. There is a disc also available with 66 programs from the book plus another 11 utilities programs and its own manual for £9.95 (£8.50 through Archive.)

• **Four podule slot backplane** for the A300 series is now available from Computerware.

• **16 bit Parallel I/O Expansion Card** from Intelligent Interfaces provides high speed data input/output on two 16 bit (each splittable into two 8 bit) TTL data ports with full facilities for handshaking. Also has four 16 bit counter/timers. Software provided for both high level and low level programmers.

• **Dual RS423 Interface Expansion Card** also from Intelligent Interfaces provides two extra RS423 serial I/O ports complete with software

support. 50 to 38,400 baud independent speed settings for input and output. Uses standard 9 pin D-type connectors.

• **OVID Toolkit Module** from P. Taylor Software. £15 for a module containing memory editor, disc editor, shift, fill, compare and find commands.

• **ArcTFS** – free-form text filing system is now First Word Plus compatible, i.e. it can output text in First Word Plus format and can also take in ASCII text from any source.

• **Plague Planet** – Text adventure game from Alpine Software, £11.95 inclusive, with over 250 locations and more than 1000 messages and many objects to find and manipulate. (Watch out for ALPS – Adventure Language Programming System – also from Alpine Software. Write your own adventure programs. £28.95 inclusive for disc and 36 page user manual. Available some time in late August.)

• **Screen Compressor** program available (by 10th August) from Stream Software. It can compress any mode to typically one third of its original size in about one third the time of an OS save routine. Price £6.95, cheques payable to "D.Jackson". **A**

Review software received...

Apart from reviews already written or in the process of being written, we have received review copies of the following software: OVID Toolkit Module.

The bits that wouldn't fit in this month's issue...

Watch out next month for reviews of SignWriter and Nick Evans' database of educational information, a discussion of the relative merits of various word-processors, an article about the development of the ArcTFS database and what the manual doesn't tell you about SpellMaster's capabilities in terms of its SWT's. **A**

Comment Column

Overseas League Table

We just thought people might be interested (as we were) to find out which parts of the world Archive reaches. Here is a league table showing the numbers of subscribers in each country.

34	Holland	3	Sweden
19	Australia	2	Hong Kong
14	Belgium	2	Israel
12	New Zealand	2	Luxemburg
10	Norway	2	Malta
9	Germany	1	Brunei
8	France	1	Denmark
7	BFPO	1	Gibraltar
7	Italy	1	Malaysia
4	Canada (+1)	1	South Africa
4	Greece	1	South Atlantic Ocean
4	Portugal	1	U S A
3	Saudi Arabia	1	United Arab Emirates
3	Singapore	1	Yugoslavia
3	Switzerland	1	Zimbabwe

Some of you might be interested in Archimedian Pen-Pals (Keyboard-Pals?). However, because of the Data Protection Act, I cannot start publishing names and addresses but if any of you, in U.K. or overseas, want to find a Pen-Pal, let us know and we'll see what we can arrange.

If you want people to make contact with you, why not use the Contact Box? It's not restricted to contacts about a particular subject.

Subscription Renewal

Your subscription number is printed on the address label (You've thrown it away already?!) and will be each month from now on. The letter at the end gives the month in which you started your subscription, so if it is a "J" (as for 95% of you) then you started in October 87 (or back-dated to October), so your subscription runs out in September. To help our administration (and our cash-flow!) and to avoid missing out on your next Archive magazine, please try to renew as early as possible during September.

What's in the mag this month?

Each month, you no doubt wonder what the next edition of Archive is going to contain – so do I!!! Because I am entirely in your hands – the reader/contributors – I don't know from month to month, what articles will come in. This month, there are very few hints and tips apart from those which Adrian has produced. Don't forget that hints and tips don't have to be complicated and clever – they can be things which are very simple and may well be obvious to some people but which you think other people may not have spotted – like the bit I put in about using *COPY.

Software for Review

I've taken to putting a list in the messages section of Eureka! of the software that we have available for review. People can then find out what is available very quickly and can offer to review it. If you are interested, keep your eyes on Eureka!

Monthly Program Discs

One or two people have asked what is on the different monthly program discs. The answer is easy for the early discs in that they just contain the program listings for the particular month's magazine. However, now that folk are sending us more programs to distribute, there's more on each disc, so the following list shows what's on each disc **extra** to the programs in the magazine. The list of what's on disc 11 may not be complete because we finish preparing the disc **after** the magazine has gone to the printers.

- 1: Help with View
- 6: RS423 patch (version 1.24)
- 7: RS423 patch (version 1.24), FastRM updated, Mandelbrot screens, Life update.
- 8: Tunes for Music Editor
- 9: Phone number database
- 10: 1WPlus printer drivers, Mode conversion programs, New Mandelbrot program.
- 11: Desktop accessories, more 1WP printer drivers, Partial renumber program, Beebug's PD file compression program, hard-disc boot program, more screen shots.

Help!

• **'The Music System'.** In the July issue of Arcon User, there was a program to convert 'The Music System' files to their THEME music file system. Has anyone thought of trying to do the same for converting the TMS to the Music Editor?

• Does anyone have experience of the **Mitsubishi AUM1371A** Autoscan (Multi-sync) monitor? Gerald Wolfe (Ontario, Canada) is having problems getting it to work in the hi-res modes.

• **GraphicWriter** – Has anyone found that it occasionally crashes out in Preview mode with an 'Abort on data transfer' message?

• Has anyone actually got **View B3.0** working on the Archimedes? We know A3.0 will work but Peter Thomas in Rutland has B3.0. Any offers?

• **Scientific American's "Computer Recreations"** articles by Dewdney are full of concepts begging to be adapted for the Archimedes. Anyone interested?

• **ADA compiler and linker.** Paul McKenna is looking for one. Is there one in the pipeline?

• **Econet.** John Reddy of the Educational Computer Centre of Ireland wants to know if anyone has details of how to implement *VIEW, *REMOTE and *NOTIFY on the Econet (Filestore Level III) using A310's. Perhaps someone could write BASIC programs to provide similar commands?

• **Automatic file copying.** Could someone write a program, asks Matthew Treagus, that searches ALL files on a disc in drive 0, copies them into directories on Econet, ADFS (Hards and floppies), RFS etc according to file type. i.e. A file marked FFA would be copied to a directory :1.\$Modules, for example?

• **BBC to Archimedes Conversions** – Now that there are various means of achieving transfer of files etc from BBC to Archimedes, there is still no ready indication of which programs will or will not run on the latter machine, either on their own or under the 6502 emulator, other than by trial and error. I am sure that many readers, would welcome an article or series of articles on how to "convert" programs and how to spot programs that cannot readily be converted. Even though more software is

now becoming available for the Archimedes, there must be a large number of BBC programs worthy of conversion. (Rob Brown)

• **First Word Plus Printer Drivers** – Has anyone done any printer drivers for First Word Plus that we could pass on to other potential users? In particular, Clive Williams is looking for one for the Epson GQ3500 laser printer. **A**

Matters Arising

• **Program Discs** – To save time when preparing program discs and shareware discs, we have had them pre-formatted by the company that supplies the discs. Unfortunately, we have noticed that the data loads more slowly than when the same discs are formatted on an actual Archimedes. The difference is only about 30%, but after Adrian has spent such a long time compressing all those screens and making them load so smoothly, it seems a shame to slow them down because of disc speed. The solution is simple. Back up the program disc onto one of your own discs (which you should do anyway for safety) and use that one, keeping the original as the backup. Or, if you prefer to use the original version, reformat it (after having taken a copy of it!!) on your own computer and backup from the copy onto the re-formatted original.

• **Duff Discs?** – Whilst on the subject of discs, we have had reports that some of the Wabash Datatech discs we have been supplying don't always format first time and need a second try. Although no one has reported any actual data errors, we thought it was better to be on the safe side and have therefore gone to a higher specification disc. To keep the price down, however, we are taking un-labelled ones. (They are, in fact, unlabelled Sony discs.)

• **Watford Digitiser owners!** Try SYS & 802D2! I'm not sure if this is what Mike Harrison (July, page 10) calls "hidden software"!

• **Quazer Passwords** – The passwords for version 1.42 are not the same as those given last month. They are: BlueIsle, Speaker, Light, Mercedes, Physics, Kojak, Electron, Plant, Barbour, Pizza and Rower. (Thanks to John Catchpole for these.)

• **The Quazer cheat** on page 8 had a line missing. After *SAVEing the code, you have to do a

Hints and Tips...

• ***COMPACT** – The User Guide says that *COMPACT “moves files around on the disc, collecting all the free space into a continuous block”. However, if you have a disc that has several gaps you will find that you need several compactions. One disc which had 5 gaps in the FS map had to be compacted eight times before the map showed only a single gap. (See review of R.A.Engineering’s Utilities, page 48.)

• ***COPY** – When copying multiple files using wildcards, you must ONLY specify the directory into which to copy and not try to specify the actual file name in any way e.g. if you want to copy files PROGA, PROGB, PROGC etc from the “BASIC” directory into the “BACKUP” directory, use *COPY \$.BASIC.PROG* \$.BACKUP.* but if you try to use *COPY \$.BASIC.PROG* \$.BACKUP.PROG* it won’t work. This is not desperately clear in the User Guide and I’ve only just worked out why 90% of my attempts to do wildcard copies have not worked!

• **CAPS LOCK** – Another thing which is obvious if you know about it is that you can set up the keyboard to print upper case normally but then go into lower case when you press <shift>. This can be done either by *Configure SHCAPS if you want it to start up in that mode (or CAPS or NOCAPS if you don’t) or by holding the shift key down and pressing and releasing the Caps Lock key.

• **CHAINing programs** – In response to our complaint in last month’s Bug or Feature that programs chaining one another cannot be called from drive 1 using the desktop, Philip Colmer of

Acorn’s Customer Support Department tells us that it is definitely a feature. There are a number of ways of avoiding the problem. Firstly, you could reference files by disc name, so we could CHAIN “:ProgDisc.\$Progs.Nextprog” or whatever. Or you could use the fact that when you run a BASIC program in drive 1 from the desktop, it generates a command something like *basic –quit “ADFS::1.\$Progs.FirstProg”. You could then read the command string with OS_GetEnv to and find out where the program has been called from. Thirdly, have a look at PC.PC from the 1.20 version of the PC emulator to see an intelligent (incredibly so!) boot program which has been carefully commented to enable you to modify it for your own use.

• **Auto-booting** – Again from Philip Colmer comes the suggestion that you should use the PC.PC program to boot up applications from within directories on the hard disc. The program uses legal OS calls instead of re-configuring. It does a *DIR and then runs the program specified. The only restriction on it is that it cannot change the system sprite size but, he points out, if programs were written properly, they would not be using the system sprite area!

• **System Devices** – You can treat devices as files, e.g. C%=OPENOUT(“NETPRINT:”). This is particularly useful in the example given on page 9 last month because you can now open a channel to the network printer, send some stuff to it when you are ready (using BPUT #C%) and when everything is done, close the ‘file’ (CLOSE#0) and it will then be printed! (This also came from Philip Colmer – Thanks PC! – Funny, you know, I always thought that PC stood for Personal Computer! Ed.)

The following Hints are from Adrian Look...

*SETTYPE QuazCode FF8. In any case, it does not seem to work on version 1.42. Can anyone help?

• **Dudley Micro Services** are moving into more spacious accommodation at 32 Osborne Road, Penn, Wolverhampton, WV4 4AY. (0902-342214)

• **Bible quotes** – Peter Howard, after reading my “graceful and apposite reference to Ecclesiastes” last month, has come up with Acts chapter 8 verses 30/31. However he reckons that, with his luck, he wouldn’t encounter a Philip but a Pharisee! **A**

• **Smooth Scrolling** – Using VDU 23,7 you can scroll the current text window up, down, left, or right. Horizontally, the picture can be scrolled by one byte, but vertically it can only be scrolled by one character cell. This can produce a smooth scroll by placing a WAIT (for the vertical sync pulse) command before the VDU 23,7. If any further delay is implemented then the picture shudders as it scrolls. This means that a slow smooth scroll cannot

Hints and Tips

be used. The answer is to redefine the screen base address manually (as we used to do on the BBC). This is done using 'OS_Word' &16.

```
DIM block% 4
:
block%?0=type
block%!1=offset
WAIT
SYS "OS_Word", &16, block%
```

"type" : when 1 – base used by VDU drivers (i.e. screen updated)

: when 2 – base used by hardware (i.e. screen displayed)

"offset" : from the address of the base of the screen buffer to the start of the screen display.

The following example program should help to make it clearer.

```
10 REM >scrolling
20
30 REM *****
40 REM *      Scrolling Screens      *
50 REM * written by Adrian Look *
60 REM *      21st July 1988      *
70 REM *****
80
90 MODE 0:OFF
100 DIM block% &10
110 count=0
120
130 PRINTTAB(3,15); "Some text"
140
150 REPEAT
160   PROCscroll(1)
170 UNTIL 0
180 END
190
200 DEFPROCscroll(speed)
210 PROCinfo
220 add=x/m
230 IF SGN(count)>0 THEN speed=-speed
240 REPEAT
250   block%?0=2
260   block%!1=add*count
270   WAIT
280   SYS "OS_Word", 22, block%
290   count+=speed
```

```
300 UNTIL add*count>x*y/m+4160 OR
                                     count<0
310 ENDPROC
330 DEFPROCinfo
340 SYS "OS_ReadModeVariable", MODE, 4
                                     TO ,,x
350 x=4-x:x=x*(2^x)*10
360 SYS "OS_ReadModeVariable", MODE, 5
                                     TO ,,y
370 y=3-y:y=y*256
380 SYS "OS_ReadModeVariable", MODE, 3
                                     TO ,,c
390 CASE c OF
400   WHEN 1 : m=8
410   WHEN 3 : m=4
420   WHEN 15 : m=2
430   WHEN 63 : m=1
440 ENDCASE
450 ENDPROC
```

• **Desktop Utilities** – The desktop is essentially a skeleton program which allows the programmer to install his/her own icons, windows, menus, commands, etc, which the DeskTop will then operate. This is done with a very clever set of FN's, for example:

```
PROCinstall("adrian")
:
DEFPROCinstall(file$)
INSTALL file$
void=EVAL("FNinstall_file_"+file$)
ENDPROC
```

This means that by EVALuating a string the DeskTop can call any file-specific function. In the case above, FNinstall_file_adrian will be called. By using this system, any command can be 'installed' or even replaced. Explaining or even listing the functions and procedures available in the DeskTop program is not really possible in the magazine. However, bearing in mind the methods used, you will find before long you can write some very useful tools for the DeskTop. It should even be possible to completely re-write the it! As an example, here is a program which will allow you to use star commands. Don't forget to *SETTYPE 'filename' FEO to indicate that the program is a desktop utility (We have included several other utilities on the program disk).

```

10 REM >star
20
30 REM *****
40 REM *Star Commands for Desk Top*
50 REM * written by Adrian Look *
60 REM * original idea D. Howlett *
70 REM *****
80
90 DEFFNinstall_file_star
100 file=OPENIN(filehandler_path$+
      ".istar"):CLOSE #file
110 IF file=0 THEN ERROR 1,"Can't
      find icon file 'istar'"
120 OSCLI("SMERGE "+filehandler_path$
      +"."+istar")
130 PROCsys_addtoiconbar("star",
      "command",&301A,icon_fgcol,
      icon_bgcol,48)
140 SYS "Wimp_ForceRedraw",-1,0,0,
      1279,100
150 =0
160
170 DEFFNaction_star
180 PROCstar_command(5,5,75,25,2,1)
190 =0
200
210 DEFPROCstar_command(x0,y0,x1,y1,
      bx,by)
220 *POINTER 0
230 VDU 26,4,28,x0,y1,x1,y0
240 y0=31-y0:y1=31-y1
250 gx=x0*16-bx*8:dx=(x1-x0)*16+bx
      *16+16
260 gy=y1*32-by*16:dy=(y0-y1)*32+by
      *32+32
270 GCOL 0,&4:RECTANGLE FILL gx-4,
      gy-4,dx+8,dy+8
280 GCOL 0,&0:RECTANGLE FILL gx,gy,
      dx,dy
290 LOCAL ERROR
300 REPEAT
310 ON ERROR LOCAL PRINT REPORT$
320 *FX 4,0
330 INPUT"*"star$
340 *FX 4,1
350 OSCLI(star$)
360 UNTIL star$=""
370 RESTORE ERROR
380 SYS "Wimp_ForceRedraw",-1,gx-4,
      gy-4,gx+dx+8,gy+dy+8
390 VDU 26,5
400 *FX 21,9
410 *POINTER
420 ENDPROC

```

N.B. (i) You will need a STAR shaped icon called 'command', saved as 'istar'. However, if you want to test the program before designing an icon, skip lines 100-120 and change "command" in line 130 to "unknown".

(ii) Because the DeskTop uses the filename of the utility as a 'seed', it important that the utility's filename be consistent with its procedure names. For example: if you rename 'star' to 'command' then the DeskTop will look for FNaction_command instead of FNaction_star, so you will get an error!

• **Making the Print Key Save** – Last month it was suggested in Neil Strong's article about making the print key print that a slight modification of the program could make it save screen shots to the disc using SCREENSAVE "\$.pic" instead of HARDCOPYFX. However, this limits you to one screen shot at a time (otherwise you will overwrite your last one). If we use the system variables and update the 'print key' program we can get it to save a screen called '\$.scr0', followed by '\$.scr1', '\$.scr2'..etc!

```

871 adr r0,command1
872 swi "OS_CLI"
920 equs "ScreenSave scr<file>"
      ; FastSave even!?
941 .command1
942 equs "SetEval file file+1"
943 EQUB 0
944 ALIGN
1020 OSCLI("SetEval file 0")
1030 END

```

If you wish to (re)set the 'file count' to n, just type:
*SETEVAL file n

• **SpellMaster Browse in Wordwise Plus** – If you have Spell-Master and Wordwise Plus you can very easily write a two line segment program which will call the browse window. i.e.

```

*BROWSE
DISPLAY

```

Thus if the program were in segment zero you could press <shift><print> while editing your text and check your spelling using the browse window. This obviously opens up all sorts of possibilities for Wordwise Plus to use Spell Master's facilities. **A**

MS-DOS Column

Ken Biddle

Welcome once again to the MS-DOS column which this month has lots of hints, tips, and advice.

Software Compatibility

First just a word about the software compatibility list which I have been compiling. The list is now getting quite long and I don't think we want the columns space taken up each month with it so I am now producing a Facts sheets with the information on. Each month in the column I will keep you informed as to any new additions to the list. If you want a facts sheets with all of the packages on at any time then just send me a stamped self addressed envelope and I will send you one.

Public Domain Software

This month was going to contain the first of a series of articles about Public Domain Software and Shareware. However all of you wonderful readers out there have been so busy sending me letters full of Queries, Hints and Tips, and general observations that there has been no space (Or time) to begin the articles. So perhaps next month. Judging by the letters I get there seems to be a lot of interest out there in this area so keep the letters rolling in.

MS-DOS Upgrades

You should only have to pay once for an upgrade to the latest version of the emulator. I have received several letters from people who paid the £15.00 upgrade fee to Acorn for the 1.09 version of the emulator. As you know Acorn then subsequently released the 1.20 version of the emulator. Well the above mentioned (Extremely Irrate) Archimedes P.C. users wrote to Acorn and complained. Acorn responded by sending them the new version. Also some of you that paid £15.00 again for the 1.20 version have been able to get their money back. Drop Acorn a line and see how you get on.

The new emulator disk contains the 1.00, 1.09, 1.20 versions of the emulator as well as an under development 1.21 version.

The main improvement with this new disk is a fully commented Boot file which kills all the unused modules so that over 500k of workspace results.

Brian Carroll writes in to say...

MS-DOS Manuals

Being unfamiliar with MS-DOS I had to buy some books on the subject. By far the best value is "MS-DOS A Quick Program Reference Guide" from Abacus/Data-Becker, J.Scheib, Jan 88, ISBN 1-55755-000-X, at just under £9.00 This is a robust hardbacked pocket book with 3 excellent indices and a standard format for succinctly displaying information on all DOS topics (up to DOS 3.3) on one or two pages. I recommend another book in the same series on GW-BASIC, H.J.Bomann, July 87, ISBN 1-55755-001-8, which I found invaluable when converting Signwriter from PC to BASIC V. There are others in the series on several of the widely-used P.C. programs such as dBASE III, Lotus 1-2-3 & Wordstar.

BASIC

ASCII source code for Microsoft Quickbasic seems to run OK under the GW-Basic interpreter. Also the Mallard BASIC supplied with the Amstrad PCW seems extremely similar (certainly much of it is identical to GW-BASIC and QuickBASIC according to the manual) so it might be possible to run PCW BASIC programs that do not use keyed files, using GW-BASIC. I have not tried this, however. *(If you try this, be sure you transfer the programs as ASCII files – the tokens for the BASIC keywords that the Amstrad PCW uses will almost certainly be different from those of GW-BASIC... Ken)*

Software for the compatibility list

Lotus 123 Paradox (Database Manager) Ballyhoo (Game) GEM (Desk Top Manager)

Hints and Tips from Karl Strickland

- To stop being prompted for the time and date each time DOS is loaded, create an AUTOEXEC.BAT file containing two returns.
- With Version 1.2, start the emulator off by using *PC.PC instead of *PC.EMULATE. You will then have no need to mess around with the *CONFIGURE commands before and after you use the emulator. This 'PC' program sets the configuration necessary to maximise the PC

memory. Your normal configuration is restored when you exit the emulator by pressing <reset>. The available PC memory now only depends on the Spritesize – if you set *Configure Spritesize 0, you get a DOS system with exactly 600k (614400 bytes as reported by CHKDSK), 504k (564160 bytes) of which is usable when MS-DOS is loaded. However, Spritesize needs to be at least 1 if you want to be able to use the desktop.

- <ctrl-C> usually stops PC programs and takes you back to DOS, but if that doesn't work, try <ctrl-break>, before rebooting (with <ctrl-alt-del>) as a last resort. To get back to Arthur, press the <reset> key on the back of the keyboard.

- When you type a file, or do a dir (without the /W option), the text sometimes flies off the screen. To avoid this, use the 'More' filter supplied on the DOS disk. e.g. Type read.me more <return> or DIR more <return>. After a page of information has been displayed, pressing any key will display the next page. The more program must be on the disk in your currently selected drive. (Alternatively, with the DIR command, you may put the pause switch on with /P. This will display one screenful at a time. e.g. DIR /P... Ken)

- The 'Sort' filter can be used in a similar way. e.g. DIR SORT will give you a sorted directory.

- If you are sick of MS-DOS always displaying the date in the US format then adding the line – country=44 to your CONFIG.SYS file will cause the date to be displayed in the U.K. format. The 44 can be changed to give other countries, and is actually the international telephone prefix number for that country.

- If you want to generate a small file (perhaps a batch file) and you haven't got a copy of edlin on your current disk, or just can't be bothered with it, then use the copy command instead. Try:

```
COPY CON: test.doc <return>
This is the first line of text in file test.doc
This is the second line and so on...
<ctrl-Z> <return>
```

The last line shows that you have finished entering the text and you will then go back to the normal MS-DOS prompt.

- Setting up your printer so that it will work with your Archimedes and the PC emulator requires some thought because by default, the Arc doesn't issue linefeeds (however you can alter this) and the emulator does. (So does the real PC – as far as I know, you can't change this – Typical of IBM!) So switch linefeeds OFF on the printer (usually with some sort of DIP switch) and type the following line on the Arc. *Configure IGNORE NO <return>

Remember to enter this line again if you ever do a Power-Up <R> reset.

- When using the PC emulator, the whole screen isn't used – just a window in the middle. This is O.K. however, if you have the Acorn colour monitor (or probably some other make as well) you can alter the size of the picture to make the window take up the whole screen. Use the knob labelled 'V.Height' on the back of the monitor – the one nearest to the scart plug. Remember to reset it again when you use your Archimedes as a real computer – i.e. as an Archimedes. *(I haven't tried the above because I don't want to re-adjust my monitor every time I use the Emulator. Seems a lot of effort when the PC emulator display is already quite adequate... Ken.)*

- In MS-DOS, it is possible to use the function keys to edit the line currently in the command buffer. This line in the buffer is the last command that you entered. This can be very useful if you want to execute several similar commands, or just repeat a single command a number of times.

F1 (or Cursor Right) transfers the next character from the buffer to the display.

F2 followed by a character, will cause all characters from the buffer up to but not including that character to be transferred to the display.

F3 transfer all the remaining characters in the buffer to the display. Useful for repeating a command.

F4 As F2 but causes characters to be deleted.

F5 Transfers the displayed lines into the buffer for further editing if desired. The line is not executed as a command.

<Insert> inserts characters into the buffer.

<Escape> leaves the buffer unchanged.

<Delete> deletes the next character in the buffer.

A Tip from Mark Barr

To copy files on a single drive under MS-DOS it is no good issuing the command `copy A:filename A:filename` as you will get the error message "Cannot copy onto itself". If you issue the command `COPY A:filename B:filename` this will, when it has read the source file, ask for a disk to be inserted in drive B: Simply put your destination disk in the drive and then press <return>. It then copies the file onto your new disk. The Arc treats a single drive as both A and B.

Batch File for Function Keys

Finally, here is a little batch file which will allow you to program the function keys to perform commands etc. Just like the *KEY command when you are using the Archimedes in native mode.

Just type the following in (Using edlin or some other text editor) and save it to disk with the name "KEY.BAT".

```
ECHO OFF
set oldprmt=%prompt%
prompt
set key=%1
if "%1"==" " set key=99
set code=32
if not %2==/i goto skip
set code=13
shift
:skip
set string=%2 %3 %4 %5 %6 %7 %8 %9
if %key%==1 set kdef=59
if %key%==2 set kdef=60
if %key%==3 set kdef=61
if %key%==4 set kdef=62
if %key%==5 set kdef=63
if %key%==6 set kdef=64
if %key%==7 set kdef=65
if %key%==8 set kdef=66
if %key%==9 set kdef=67
if %key%==10 set kdef=68
if %key%==99 goto help
goto found
:help
ECHO KEY Syntax rules.
ECHO
ECHO " key num /i <definition> "
ECHO.
ECHO num is the number of the
function key to be defined,
```

```
ECHO      This must be in range 1-10
ECHO /i    This should be included
           if you wish the command
           to execute immediately.

ECHO
goto done
:found
set key=
prompt
$e[0;%kdef%;"%string%";%code%p
ECHO ON
:done
ECHO OFF
set code=
set key=
set string=
prompt %oldprmt%
set oldprmt=
ECHO ON
```

Well that's it for this month. Don't forget, if you want a facts sheet, send me postage and an envelope and it's all yours. I'm going to try and set up a Public domain disk for the PC emulator and distribute it to anyone that wants it. Should anyone have anything that they wish to contribute and get their name in lights on the disk, please send it in.

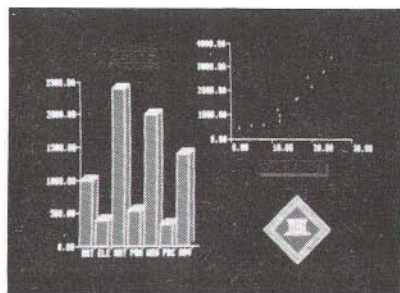
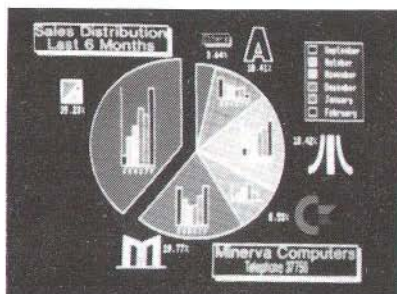
Last but not least I must offer my humble apologies to everyone that tried to set up a Ramdrive using my instructions in a previous article. I'm afraid a mistake crept in. (Lack of co-ordination between my brain and my fingers.) I told you to enter the command "system" and in fact it should have been "device". Oops! Many thanks for all the people who wrote in and told me of my mistake. So just to set the record straight here it is again:

```
EDLIN CONFIG.SYS <return>
I <return>
DEVICE=RAMDRIVE.SYS xxx 512 64
<return>
CTRL-Z <return>
E <return>
```

Where xxx is the size of the ramdisk in kbytes. You now need to re-boot the machine (ctrl-alt-del). When the disk is Re-booted it will show which drive accesses the ramdrive. One final bit of info - as you can see from the additions to the software list this month, GEM seems to work just fine and that includes using it with the mouse. I will try and include the mouse driver on the Public Domain disk when it is finished. **A**

GAMMAPLOT

Presentation Graphics with a full Art package
to unleash the power of the Archimedes



Seeing is believing!

GammaPlot is an extensive chart / graph plotting program with a full art package included. It allows even a novice to produce stunning charts and drawings on one screen. Several charts may be placed on one sheet, in any position with text or drawings to make the data easily understood. GammaPlot supports six main graph and chart types of line, scatter, pie, histogram, 3D histogram and text only charts. Segments may be highlighted or percentages, values and labels switched on. Statistics and a full Slide Show are also included.

Choose from the text styles supplied or any of the native Archimedes 'pretty fonts' or even define your own, and any of these can be enlarged or reduced. Various shapes, patterns and borders are supplied with a full 256 colour range and a flood fill option. A full zoom facility allows editing of individual pixels. Various pen types and spray cans are provided to draw pictures or add signatures. Block options include move, copy and squash.

Import digitised pictures or company logos. Print facilities include colour output. Data entry is via the keyboard or any Delta, Gamma or Sigma file or most other graphics programs. Screens may be saved and merged automatically with wordprocessors like 1st Word Plus.

So extensive is GammaPlot that recent presentations to educational sectors show that many schools are buying GammaPlot for the art package alone!

AFFORDABLE



EXCELLENCE



MINERVA

COMPUTER SOFTWARE AND SYSTEMS SPECIALISTS

69 SIDWELL STREET · EXETER EX4 6PH · TELEPHONE EXETER 0392 437756

BASIC V Utilities Forum

Clifford Hoggarth

Faster Graphics – SYS, VDU or...?

Ian Smith's article on using SYS calls prompted Roger Darlington to write questioning whether using SYS "OS_Plot" is really faster than using VDU calls and pointing out that BASIC's POINT command was faster than both anyway. Now unfortunately I could not find my copy of the original article on which the timings were based (I know it's under one of those piles of paper on the floor, but which one?!) so I've had to do my own investigations using the listing opposite. The loop was run with line 60 empty to establish the value for empty_time. Thereafter each command was added as line 60 and the times produced were noted. The results for plotting a point at (0,0) on a mode 12 screen are:

Command	Time (seconds)
VDU 25,69,0;0;	4.35
SYS "OS_Plot",69,0,0	24.45
SYS &45,69,0,0	2.81
SYS OS_Plot,69,0,0 {OS_Plot=&45}	2.79
SYS O%,69,0,0 {O%=&45}	2.73
PLOT 69,0,0	1.98
POINT 0,0	1.78

At first glance, the VDU calls appear quicker than using SYS "OS_Plot", but as can be clearly seen, if the SYS call is made by number rather than name then SYS "OS_Plot" is the faster of the two, as explained in last month's column.

Assuming Arthur's VDU sequences are the same as those for the BBC, it can be seen that 6 calls are needed for the VDU command to pass the necessary data one byte at a time, whereas the SYS "OS_Plot" routine, although having the overhead of saving BASIC's register values, only requires a single call. It seems likely that the VDU drivers eventually perform the plotting using SWI "OS_Plot" anyway. Having said all this, from BASIC the fastest method is to use the built-in commands. A little thought shows that this is quite logical since (i) less interpretation is necessary, there being only two

parameters required for BASIC's POINT command, which is fewer than are required for any of the O.S. methods and (ii) the actual O.S. call can be optimised, storing only those registers necessary, whereas a SYS call from BASIC must store all the registers as the interpreter does not know which will be corrupted on returning from Arthur.

So, in conclusion, for fast graphics use the built-in BASIC commands. If a suitable command is not available, e.g. for sprite manipulation, then it is quickest to use the relevant SYS calls, but by number rather than by name.


PRINT USING

Roger also commented on the lack of a PRINT USING command in BBC BASIC. It does seem a little strange that Acorn have not added one – it cannot possibly be due to a lack of space! Now past issues of Archive have contained routines for aligning decimal points, but what about a general routine along the lines of PROCprint_using (X,"60#60") etc. If you have any ideas on this, please send them in to me care of Archive.

Listing – To time various commands

```
10 empty_time=17
20 MODE 12
30 N%=10000
40 TIME=0
50 FOR C%=1 TO N%
60   REM Command to be timed
70   NEXT
80   T%=TIME
90   T%-=empty_time
100 PRINT T%/100 "seconds"
110 END
```

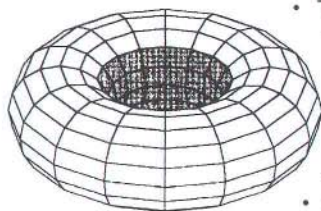
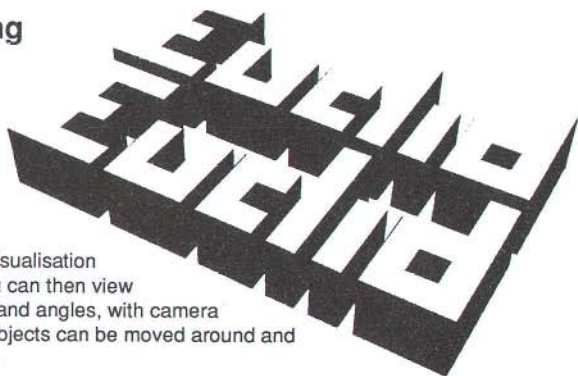
Contact Box

- Broadcasting and Digital Sound Recording – if interested, contact Ned Abell, Room 329, BBC Pebble Mill, Birmingham, B5 7QQ. 

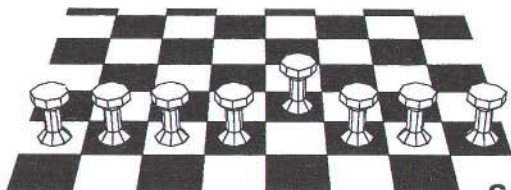


Euclid

is a fully programmable 3-D modelling and animation system. It allows instant visualisation of scenes in 3-D which you can then view from a variety of positions and angles, with camera control like a TV studio. Objects can be moved around and constructed live on screen.



- Typical drawing times of 1/5th second for full screen
- Intersecting and concave surface polygons
- Line or surface drawing
- Full hidden surface removal
- Instant scene and object transformations
- Works in any graphics mode
- Multiple use of solid definitions in one scene
- Orthogonal or perspective projections
- Conversion to and from DXF format
- Supports a number of colour printers at full resolution



Cost: £45 (including VAT and P&P)



Order Form

Please send me _____ copy/copies of Euclid.

I enclose a cheque/PO for £ _____ (made payable to Ace Computing).

Name: _____

Address: _____

Date: _____ Signed: _____

Please complete and return to: Ace Computing, 27 Victoria Road, Cambridge CB4 3BW

Languages Corner – Pascal

David Wild

I bought Acornsoft Pascal even before I took delivery of my machine, so that I could read all about the things I would be able to do with the new system. Unfortunately the manual was rather disappointing. There are pages and pages of examples relating to the error messages and many of these examples contain their own mistakes, but hardly anything about the use of the system. This applies especially to the extensions Acorn have provided.

When my machine arrived I tried out the various facilities and found that the actual system was excellent – Acorn had done a very good job of hiding the strong points of their own program!

There is, for instance, an “include” facility which allows the compiler to draw parts of the program from files rather than needing to bring them in to the source file first. This may seem to be only a slight benefit but it has the extremely important effect of eliminating the editing of source files if you change one of the included procedures or functions. Such a facility has been available in Turbo Pascal for a long time but what Acorn do not mention is that the inclusions can be nested. You can include a file which includes a file which includes a file. I presume that there must be a limit somewhere, but it certainly works to four levels and more than that would be greedy!

There are several other minor, but useful, extensions to do with such things as binary and hex constants, the ability to ‘and’ and ‘or’ integers, and the making available the address and size of variables. In file reset and rewrite statements there is provision for checking that the file has been opened successfully for the access required.

Provision is made for the use of ‘inline’ machine code statements and, although Acorn say that this should rarely be necessary, it is extremely important in the extending of Pascal to use the ARM facilities to the full.

The most important extension of all, however, is the provision of ‘modules’ which can be compiled separately and then attached to a program by the linker. We are already seeing advertisers offering packages of modules to perform such functions as

graphics, and no doubt there will be many more as time passes.

Included in the modules part of the system is provision for ‘static’ variables, which will keep a given value until altered explicitly. This is extremely significant, as we can now write procedures or functions which “know” whether or not they have been called before without the need for global variables, or any other action in the calling program.

Unfortunately the modules part of the manual occupies only three pages in spite of its being the most important part of the extensions. It is to be hoped that someone will write a guide to modules, leading the programmer through the difficulties and opportunities.

I have found very few bugs and none of them more than irritating, in the program, although there are probably several which haven’t cropped up yet.

On the whole I feel that the new Pascal is an exciting package about which there is a lot to be learned. It generates machine code and, after linking, can be run on any Archimedes with sufficient memory for the task to be done. I have not yet done enough tests to find how to decide when a program will run on a 305 or 310, but I suppose that this information will start trickling through. There is no mention of royalties, so it can be used for the generation of programs for sale. The main improvement needed is a decent manual because without one, many opportunities may be missed.

ISO Pascal Extensions

from Ian Smith & Leslie Wiggins

Now that Acorn Pascal includes the idea of separately compiled modules to be linked to other programs it is only to be expected that there will be programmers offering pre-written modules to extend the system.

Smith & Wiggins are leading the field with a set of seven modules which can be attached to your own programs and so save you the work of researching and writing your own. The modules are “General”, which provides many of the interfaces with the operating system which are taken for granted in

BASIC but which are completely missing in Pascal as supplied, "Graphics", "Sprite", "String", and "Wimp", whose functions are obvious, together with "Glib" and "Transform" which are very much concerned with line graphics and windows.

In addition to the modules, the disc contains sample programs using the various facilities and full source code is provided for these programs so that you can see how the modules are used and even make your own modifications to help your understanding.

These sample programs are very impressive and really do make good use of the extensions. The weakest part of the whole package is the manual. In it, the authors have tried to provide documentation for their package and also explain how you might write your own modules. There are 22 pages prepared, I suspect, using First Word Plus, and copied on one side of the paper only.

This causes problems of clarity because of the density of text on the page. There are times when it is difficult to see where one sub-topic finishes and the next one starts. The manual would be very much improved by being considered by a typographer.

In the "Glib" and "Transform" modules there are some rather unfamiliar ideas, particularly relating

to "viewports" and these could have done with rather more explanation.

I can sympathise very strongly with the authors who have to reckon with a fairly limited market but the appearance of the manual may deter some people who would find the modules extremely useful.

There is no mention of any restrictions on incorporating these modules in fully-linked programs for sale, although there is a very reasonable plea to programmers to give their friends the address of the authors rather than copies of the modules.

I can see a problem which the authors will face in the near future as the idea of modules becomes more popular. I have written some procedures which could be included in modules for sale, but they would be even better if they included some of the procedures from this package. Presumably it will be necessary to sell two versions of such modules, one for those who have bought the Smith & Wiggins package and an inferior one for those who have not.

This is a package which can be thoroughly recommended. The time saved in not having to write some of the operating system interfaces more than justifies the purchase price of £30, and the graphics and wimp extensions are even better value. **A**

Archimedes

5.25 Disk Interface

Fit a 5.25 Drive to your Machine the safe and easy way using our
NEW DISK INTERFACE

- Available Now
- No soldering
- Supports 4 Drives
- Low Power Consumption
- Fully Buffered
- Easy to Fit
- All Cables Supplied
- Full Instructions

In Stock Now and Despatched the same day on receipt of your
Cheque / Postal Order for £27.95 inc P & P.

Dudley Micro Services

32 Osborne Road, Penn, Wolverhampton, WV4 4AY.

Telephone: 0902-342214 / 334315

Hardware Corner

Brian Cowan

Podule News

There are some super goodies coming from Intelligent Interfaces. These are the people who brought you the IEEE488 interface podule. Three podules should be available shortly.

For those fed up with the Archimedes serial port and its problems, there is a **twin RS423 podule**. This uses a completely different serial chip which has no faults. Together with its driver software this should make serial communication a doddle. Also on the podule is a limited amount of parallel I/O in the form of the old familiar (to BBC buffs) 6522 VIA.

Then there is a **16 bit parallel I/O podule**. This has two full 16 bit ports either of which may be alternatively configured either as two eight bit ports. There are handshake lines provided and the pinouts correspond to a double User Port. For those who are interested in 16 bit parallel I/O, the choice is between this podule and the one from RESOURCE. So far I have seen neither but I will give more information a.s.a.p.

Finally, Intelligent Interfaces are working on a **12 bit analogue podule**. This should have eight multiplexed analogue inputs and it should convert at a rate in excess of 100 kHz. There will be software selected sensitivity setting and I understand that a lot of effort is going into the analogue side of the design.

Incidentally, owners of the **Acorn I/O podule** have an extra serial port in the form of the MIDI hardware. Two ICs are needed, a 6850 serial chip and a 6N138 opto-coupler, to be placed in the empty sockets on the board, and a connector is required to the serial line. Some enterprising programmer should consider writing some software to drive this port.

There is the **Buffer podule** available from SGB Computer Services allowing one podule to be mounted externally, and the anticipated expansion box that will accommodate up to 5 such podules. However note that not all podules may be mounted externally, only

Simple podules and External podules but this excludes the Acorn I/O podule which is really a MEMC podule.

For the soldering iron brigade there are the **do-it-yourself podules** produced by Atomwide. A certain amount of line buffering and latching is provided together with address decoding, but after that it is up to you. I have not actually seen one of these podules yet, so I don't know how much documentation is provided. However when I do get my hands on one I hope we shall run some projects using them.

Podule fans will have greeted with joy the announcement of the **four slot backplane** for the 300 machines from Computerware. However, a word of warning concerning power supply ratings. It seems that the power supply in the 300 machines could not cope if all four podules drew their full entitlement of current. This is probably unlikely to occur and if it did, the switched mode p.s.u. should simply cut out with no permanent damage to hardware, but you've been warned!

I²C Serial Bus

I received a letter from Steve Church of Norwich in which he asked about the I²C bus on the Archimedes. For those who don't know what this is, the I²C bus is a two line serial bus that is patented by Phillips. Various chips such as ROMs, RAMs, real time clocks, display drivers, etc, etc support the bus and it is used in the Archimedes; its real time clock/CMOS RAM is such a chip.

The input output controller (IOC) functions as the master through its external control lines C0 and C1. These act as the I²C bus serial data and serial clock lines. Within the Archimedes, it is only the RTC/CMOS RAM chip which uses the bus. However, the two bus lines are brought out to the podule bus to pins 20 and 19 of row C. Thus, external devices may be hung on this two line bus; the ultimate in hardware simplicity.

The question then arises as to the machine firmware support for the bus. Clearly there is something there for communication with the RTC/CMOS RAM, but are there any SWIs or operating system "hooks" that may be used? Unfortunately, Acorn were not able to offer much help in that line except to say that there is support for I²C in Arthur version 2. Anyone with any more information, please let us know!

The Memory Controller – MEMC

The four components of the Archimedes chip set are designed to go together as a unit. For this reason there are functions squeezed into each chip which are unexpected considering the chip's name. Thus, the MEMC generates the various timing signals for the system, deriving them from the 8 MHz clock. Another surprising feature of the MEMC is that it does not connect to the data bus. It connects only to the address bus together with a number of control lines. So, in communicating with the MEMC, the address written must contain the "data". In practice it is much more convenient to use the SWI provided for that purpose: OS_UpdateMEMC. Some aspects of this will be discussed below.

Logical and Physical Space

The main task of the MEMC is to look after the Archimedes memory. The 26 address lines of the ARM allow it to address 2^{26} or 64M locations. By convention, the addresses refer to byte-wide locations, so since the ARM has a 32 bit data bus, we are talking about 16M 32bit words. For whole word transactions, the lowest two lines of the address bus will be unused. The upper and lower 32 Mbytes of memory are treated differently. The upper half contains all physical devices, ROM, RAM and all I/O devices. However, in normal usage, not much of this space can be accessed; only the ROM can be read. To understand how to read and write RAM, we must start with the concept of logical RAM.

The maximum of 4 Mbytes of physical RAM is mapped onto the lower 32 Mbytes of address space by the MEMC. The lower half of the machine memory address space is referred to as logical RAM space. There is no restriction on reading and writing to this area and the logical-to-physical translator in the MEMC makes the conversion. The user is therefore unaware of the physical location of stored data; it is only the logical address that is known.

Protection Modes

In fact there are three levels of access to machine memory. These are known as protection modes and the "normal usage" referred to above is called User mode. The other two modes are the Operating system mode and the Supervisor mode. Normal operation, including running BASIC, occurs in User mode. This explains why you can not access

the Acorn I/O podule functions directly by peeking and poking to it as you could (illegally) on the old BBC machines. It is a function of the MEMC to monitor all address requests and to test their legality considering the current protection mode.

The supervisor modes are the most privileged, where the entire memory can be accessed. However, this must be done from machine code only. Within BASIC that means having a patch of assembler which drives the system into Supervisor mode, does the peeking and poking and then returns the system to User mode. Clearly it is unwise to do this unless it is vital, perhaps for speed of access. It is usually preferable to use the OSBytes, SWIs and other MOS routines provided.

MEMC Control Register

The MEMC control register controls many functions of the MEMC. However, since this chip is a write only device, for convenience, a copy of its current state is maintained in RAM. At any time this may be read, to examine the current state. The MEMC state may be altered by writing to the software copy and updating the MEMC from that.

When the machine is turned on or following a break condition, the MEMC control register is reset to its default value from the operating system. Some examples of what the control register actually controls are given below. User access to the MEMC control register is conveniently provided by SWI 26, "OS_UpdateMEMC". This SWI takes two arguments, R0 and R1. The first argument R0 contains the new bits to be replaced in the register. The second argument R1 is a field mask indicating which bits of the register may be changed. In this way additions to the register may be made without having to rewrite its entire contents each time. After calling this SWI, the state of the register is deposited in register R0 so that it may be read.

Speeding Up the ROMs

It is the MEMC which controls the speed of access to the ROMs. The MEMC allows for two areas of ROM each with its own access speed. Thus "regular" cheap ROMs may be used for most functions and the more expensive high speed ROMs for those functions requiring high speeds. The Archimedes, however, uses a single area for ROM (supporting a maximum capacity of 0.5

Mbyte). The access speed for ROMs is contained in the 6th and 7th bits of the register (counting from zero). The values and speeds are: 00 – 450 nS (default), 01 – 325 nS and 10 – 200 nS. Mike Harrison has pointed out that the ROMs on many machines can be run faster, at the full 8 MHz. He has provided a BASIC program called "speed" on the magazine disc for testing ROMs at the higher speed. I have tested my 310 machines and they all passed. However my 440 unfortunately failed – I think this has older ROMs, although they are all OS 1.2. If your machine passes the test, you can re-program the MEMC for faster access using (from BASIC)

```
SYS"OS_UpdateMEMC", 64, 64
```

Return to normal speed is by

```
SYS"OS_UpdateMEMC", , 64
```

Also, pressing <break> will return to normal speed.

The argument 64 is used since this corresponds to binary 1000000. The second argument of the SWI then tells it that it is bit 6 that must be changed. To speed up the ROMs, this bit is set to one while to return to normal it is set to zero. In both cases bit number 7 remains unchanged at zero. The result of this speed-up is something like a 25% increase in graphics and ROM BASIC speed. In fact ROM BASIC then runs as fast as RAM BASIC. However, Mike points out that running the ROMs faster could be unreliable, so it should not be used in commercial software, only for private use. NO responsibility will be accepted for lost data etc. resulting from the use of this technique! Note, however, that no damage can be done – in the worst case, you will have to reset the machine. Mike suggests that you run the test program when the machine has been running for a few hours to allow it to warm up – ROM speed decreases with increasing temperature.

Distinguishing Machine Types

The MEMC treats the physical RAM as a set of 128 sequential pages. Similarly the logical RAM is treated as a set of (many more) sequential pages. The logical-to-physical address converter then works by relating logical page numbers to physical page numbers. The physical page image appears repeatedly in logical pages with a period of 128. For machines with up to 1 Mbyte of RAM the size of the page is 8 kbytes, while machines with 4 Mbytes of

RAM have a 32 Kbyte page size. As pointed out in Archive 1.8, examining the page size is a way of distinguishing between 310 and 440 machines. The page size is stored in the MEMC control register in bits 2 and 3. The value 01 corresponds to 8 kbytes and 11 to 32 kbytes. To interrogate the control register the SYS function may be used with "TO" to return the contents. (The ins and outs of the SYS command were explained most clearly in Ian Smith's article in Archive 1.9.) Thus if you type in

```
SYS"OS_UpdateMEMC" TO X
```

then the contents of the control register will be placed in the variable X. Then using PRINT ~X will return the hex value which may be decomposed into the relevant bits. I obtained

```
310          440
X = &36E0D84  X = &36E0D8C
```

It is the last digit which contains what we want:

```
4 = 0100 C = 1100.
```

Thus in my 310, bits 3 and 2 contain 01 which corresponds to a page size of 8 kbytes, while in my 440 bits 3 and 2 contain 11, a page size of 32 kbytes.

That's all for now folks! Unfortunately space (and time) do not allow me to go into questions about ROM. This will be done in a future article. **A**

Small Ad's

- PC Emulator version 1.20, new, boxed £60.00, Minerva Deltabase, new, boxed £12.00. Alan Phillips, Wigan, 0942-717454
- Viglen 40/80T double-sided dis drive, as new, £70. Phone (0324) 558692.
- Logistix software, new unused, £75.00. Also, Zarch, £5.00. Phone (0633) 893957.
- Arctist, EMR Soundsynth & Creations discs, Fortran 77 for sale, phone Paul on 041-777-6608.
- Archimedes 310 colour + 2nd Drive and Acorn monitor + software. Phone 01-310-7162 (evenings) or 01-526-1578 (day). Buyer collects (S.E.2) £800.
- Archimedes 310M Colour, little used, all original packing etc, complete with GraphicWriter, PRM and ArcWriter! As new, quick sale £900. Phone (0493) 652418. **A**

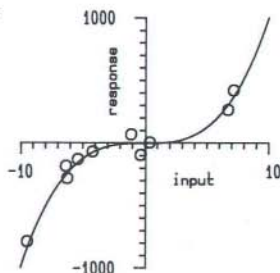
New software for the Archimedes

FORTRAN 77 Libraries

£ 49.50

A set of 3 libraries for FORTRAN 77, as reviewed in Archive (May/June).

VDU Graphics Library (for PLOT, CIRCLE, etc). Graphics Extension Library for sophisticated graphs, histograms, pie-charts, etc, (on standard and multi-sync monitors, and PLOTmate and HPGL plotters). Operating System Interface Library for access to *-commands, clocks, ADVAL, INKEY, OSBYTE, OSFILE, etc, all SWI calls, and user-written machine-code routines.



The XED Text-File Editor

£ 19.50

The ideal editor for Pascal, FORTRAN and C programs, command files, etc. Powerful full screen editor using 40, 80 or 132 column screen modes and all ASCII characters 0 - 255, plus comprehensive mainframe-style editing language including sophisticated file input/output and facility to execute command files. Full access to *-commands plus ability to compile and run programs, enter BASIC, etc, while editing.

Printer Module

£ 12.50

Tired of having to wait while files are printed? Load our Printer Module from disc and your Archimedes instantly has a very versatile printer buffer (size 1 Kbyte up to maximum available RAM) that allows you to carry on programming while your printer is churning out paper. Module also has very fast text and graphics screen dumps for Epson-FX compatible printers, and commands to send text files to the printer.

Each package is supplied on disc and includes a comprehensive user guide. FORTRAN Libraries and Xed Text-File Editor both require machine with at least 1 Mbyte of RAM.

Please write for further details, or send your order with a cheque or P.O. payable to 'CCD Computer Services'. No credit cards please. All prices fully inclusive. Official orders from Universities, Govt. Labs, etc welcome.

CCD Computer Services

71 Marlborough Park Avenue, SIDCUP, Kent DA15 9DL.
Tel. 01-302 5427.

Using Hard Discs

Brian Cowan et al

Many of our readers are interested in the question of hard discs for the Archimedes. Some argue that for "serious" use a winchester is vital, while others view the thing with trepidation, living in fear of the dreaded "head crash". Then there are those enthusiasts who wish to install or upgrade their own systems. The May issue of *Archive* carried an article by Alan Glover on fitting the Acorn hard disc upgrade to a 300 machine (some extra hints are included below). This month we shall look at some of the other questions mentioned above.

Way back in March, Alex de Vries speculated on some aspects of hard discs and the way the system is implemented on the Archimedes. This spurred others (including myself) to dig deeper. I am grateful to Mike Harrison, Gerald Fitton and D.J. Morley for sending in information, to Hitachi for giving information on the hard disc controller chip and various winchester suppliers for information on the current state of the hard disc market.

Interface Standards

The hard disc interface used by the Archimedes is not SCSI, it is the ST506 standard as mentioned in *Archive* issue seven. Bare winchesters, without an interface card, come with a number of pinout/interface connections and ST506/412 is the most common. On the 400 series machines what would be the interface card is incorporated on the main board; on the 300 this is a podule.

The SCSI standard refers to the Small Computer System Interface. This is a local I/O bus which is implemented on various microcomputer systems. Some do indeed communicate with hard discs along this bus; not the old BBCs however—they used the Acorn 1 MHz bus. Acorn are planning a SCSI interface podule for CD ROMs etc and for this reason they have reserved SWIs and IDs for the SCSI. This is nothing to do with the hard disc interface! Use of the SCSI for the hard disc would be slower and more expensive. The Acorn system hard disc controller chip, the Hitachi HD63463P8, connects to the winchester and directly to the Archimedes I/O bus which is 16 bits wide (the SCSI is 8 bits).

Data transfer is therefore faster and, furthermore, the controller operates at the full 8 MHz. The winchester in my 440 is a Tandon model TM362 which has a 28 mS access time—quite fast by normal hard disc standards.

How many Winchester's?

If you look at the main board/podule connections to the winchester, you will see three plugs. (On the podule, the

third plug may be absent but there is a space for it.) There is one 34 way plug and two of 20 ways. The winchester drive uses two of these, it connects to the 34 way and one 20 way plug. The 20 way cable carries the data while the 34 way cable contains the control lines.

The interface hardware and the Archimedes software will support two hard discs. Each disc has its own data cable while the control cable is shared; this is "daisy chained". So, space permitting, it is a simple matter to add a second hard disc drive; an extra interface card is not required. The system must then be *Configured for two hard discs which are accessed as drives 4 and 5.

Hard Disc Capacity

There is an upper limit on the capacity of hard disc drives that may be connected to the machine. This is set by the disc interface chip. The limit is 64 Mbytes per drive, which corresponds to 1024 cylinders and eight heads. However, it appears that the largest 3.5 inch winchester available has a capacity of 40 Mbytes; bigger drives would have to be mounted external to the machine.

Fitting the Acorn Hard Disc

Last month's article by Alan Glover covered this in some detail. Of course the installation should really be done by an approved dealer but this may not always be convenient. I have installed Acorn winchesters in two machines so far following Alan's instructions with no problems. However, I did not remove the plastic front from the machine. I remembered the difficulties I had following a 305/310 memory upgrade. When replacing the fascia it must be aligned correctly in relation to the floppy disc drive, otherwise there can be problems—not to mention getting the floppy drive push button in position without damaging it!

It is possible to fit the hard disc upgrade without removing the front. The only problem is with the l.e.d. which must be inserted into a hole below the power l.e.d. on the front panel. You can see where it must go but you can not get at it; there is room for neither fingers nor pliers. The answer is to use a surgical clamp/seizer, preferably with curved jaws. With this you can hold the l.e.d. quite firmly while it is manoeuvred into place. You can obtain this surgeon's tool from reputable electronics suppliers such as RS Components or Farnell.

Formatting a New Winchester

If you are lucky enough to have a 440 machine or a 305/310 with a dealer-installed hard disc upgrade, the hard disc should contain a winchester formatting program.

Clearly it is not a lot of use there and it should be transferred to floppy as soon as possible. Ordinarily you would use this if the winchester had become corrupted.

The formatter operates in an intelligent manner; initially it interrogates the hard disc as to its size and structure. It finds out such things as the number of sectors per track, the number of heads and the number of cylinders of the drive. It then asks if the user wishes the disc formatted as such before commencing operation. Thus the formatting program should support any add-on winchesters.

Other Options

If you have an Acorn winchester system, either a 440 or a 300 with winchester podule, you may add a second hard disc drive as described above. It may be possible to squeeze a 10 or 20 Mbyte 3.5 inch drive into the microcomputer case (check loading of power lines first), but one of larger capacity will have to be mounted externally, with its own PSU.

It is unfortunate that Acorn do not sell the podule card separately; one is forced to have the 20 Mbyte drive. However Watford Electronics should be providing a solution. They will be bringing out a winchester podule which will be sold with a 20 or 40 Mbyte drive and also separately for use with larger capacity units. The price has yet to be announced. It should also have the possibility of being used with ex-BBC drives as well as higher capacity units.

It is not possible to use the ex-BBC drives on the 1MHz bus of the Acorn I/O podule because of the lack of firmware support. You would have to write your own ADFS, or severely hack Acorn's – the ADFS can (currently) only talk to a Hitachi controller.

Winchester Do's and Don'ts

It is vital that care is exercised in the use of hard disc drives. D.J.Morley has sent in his ten points for hard disc users. Follow these and you can't go wrong.....

Ten Points for Hard Disc Users

D J Morley

1. Treat the drive with respect. It is quite robust in normal use but contains delicate mechanical components which can be damaged if the drive is jolted.
2. Try to remember always to park the drive's heads before switching off. This can be achieved by typing *BYE at the BASIC or Arthur prompt or by clicking <menu> on the hard disc icon in the desktop, followed by clicking <select> on the small window titled "Winnie". Do this immediately before switching off the power as any subsequent hard disc accesses will naturally unpark the heads again.

3. If you forget to park the heads or if you suffer a power failure whilst using the computer don't panic. Provided you do not jolt the drive, you can turn on the power again and it will function properly. Of course, you should never attempt to move a hard disc unless the heads are parked. You should never turn off the power whilst the disc access light is lit (this also applies to floppy discs) because if the drive happens to be writing to the disc it may be left in an inconsistent state, resulting in a broken directory if it was updating a catalogue at the time you removed the power.
4. Never, but never attempt to dismantle a hard disc drive. The magnetic discs themselves spin at high speed inside a sealed metal container in a totally dust-free environment. Whereas in a floppy disc drive, the heads press against the disc surface and actually deform it, in a hard disc drive the heads do not touch the magnetic surface of the rotating discs but are aerodynamically shaped to "fly" just away from the surface. If they should ever touch, severe damage will be sustained by both the head and the disc surface. A particle of dust is larger than this gap and would prove fatal to the drive if allowed to enter.
5. When you receive your 440 you will not be provided with a Welcome disc as all the Welcome files are already installed on the hard disc. There is also a file in the Library directory called WForm. This is a program which will enable you to reformat the hard disc in the rare event of its being corrupted. Of course, if the disc is corrupted you might not be able to read WForm from it so you are strongly recommended to copy the file onto a floppy disc as soon as possible.
6. Take regular backups. Copy all the files on to floppies at regular intervals. I recommend that you keep a set of floppy discs specifically for this purpose, clearly labelled and kept separate from your other floppies. A useful utility that someone might like to write and submit to Archive would be one which reads each sector in turn from the hard disc and copies it to a floppy, requesting that the user inserts another floppy when the last one is full. The utility would have to be able to work the other way round as well, to enable the hard disc to be recreated from a set of floppy discs.
7. Make full use of the hierarchical directory structure provided by ADFS. When using floppy discs you tend to use a different disc for each application – you might have a disc for BASIC programs, a disc for 1st Word Plus documents, a disc for Artisan files, and so on. The tendency is to put most of the files in the root directory of the floppy disc and only group files in a subdirectory if you need to. With a hard disc, you need to be more methodical. Ideally the root directory

should contain only directories (and a file called !Boot if you need one). Working out a directory structure requires some thought and careful planning and anyone who currently uses floppy discs and thinks he might buy a hard disc in the future would do well to plan ahead. I treat my floppies like low capacity hard discs. By this I mean that I only ever put directories in the root directory. So, if I have a floppy disc containing BASIC programs I am currently working on, I put them in directory \$.Basic instead of directory \$.

8. Make full use of the system variables Run\$Path and File\$Path. When FileSwitch is called upon to run a file it looks at the value of Run\$Path in order to determine where it should try to look for the file. The default value of Run\$Path is, %, which means that FileSwitch should look first in the current directory for the file. If it fails to find the file there then it should look in the currently selected library directory. Only if it fails to find the file there will it report an error. By using *SET to change the value of Run\$Path, you can tell FileSwitch to search other directories (or other discs, or even other filing systems). For example, *SET Run\$Path %,.,\$.Modules,\$.Utils. would add two more places for FileSwitch to search.
9. Occasionally verify the hard disc by typing *VERIFY 4. This check will take a couple of minutes to check every sector on the hard disc. If the process reports an error you would need to reformat the hard disc, having first copied its contents to floppies. Similarly, you should occasionally compact the disc so that the disc space which is released when files are deleted is kept together in a contiguous block rather than being fragmented all over the disc. Failure to do this can result in your being unable to save a file of a given size even though there is enough free space on the disc, because the free space is split into small fragments and none of them is large enough. A record of the areas of free space on the disc is kept in the Free Space Map which is held in the first few sectors on the disc. The Free Space Map can only hold a finite number of entries and you will receive a warning to compact the disc if it becomes filled to near its limit.
10. Finally, don't be afraid to use the hard disc. Treat it carefully and it will serve you well. Its superior speed of access and its much increased capacity over floppy discs make it a very useful addition to any computer.

Now let's hear from Gerald Fitton who concentrates more on the software techniques...

Use of Hard Discs

Gerald Fitton

There are two different strategies adopted by software writers. In the period "Before Arc" I, and many others, used the Currently Selected Directory (CSD) strategy (on my Master), but now I would suggest that the User Root Directory (URD) strategy (which I used first on Econet) is more appropriate and, incidentally, better. For either, the root directory of the hard disc, the \$ directory, should contain directories and precious little else.

Each application should have a directory allocated to it: that directory contains all the files peculiar to that application. It would not contain, for example, the floating point emulator module, which should be in a directory called "Modules", however, it would contain both programs and data files for the application.

When you use a floppy disc there is a !BOOT file. Transfer this to the hard disc's Library directory giving it a new meaningful name such as Art (for Artisan) or PipeDream (for the software of the same name). If you do this then, from anywhere on the disc you can type *Art (or whatever) and that "!BOOT" file will be run. The !BOOT file (now called something else) should QUIT, RMLOAD any modules or do any other chores of this kind but, if you are using the URD strategy, it should always set the URD to the directory that contains the programs relevant to the application.

Using this strategy, and the Artisan example, there will be a directory called Artisan and this will have been made the URD when you typed *ART. Artisan contains all the files relevant to the use of that software, both program files (supplied by Clares) and data files (your pictures). If the software has been written in the URD style, then when a data file is called up by the program, it will call it using the ampersand, "&", and so find it as a file within the URD.

The CSD strategy is similar except that data files are called by using a sequence of *DIR commands such as *DIR ^,DataFiles.Mydata to move around the directory structure, making each directory the Currently Selected Directory, even if only on a temporary basis. This technique is a bit harder on the programmer and can cause problems if the suite of programs gets complicated.

Finally, tidy everything up after a session, getting rid of junk and backing up (using *COPY) anything important to a floppy disc, follow this with *COMPACT and *MAP until your disc has only one large contiguous free space, and finally type *BYE, to park the heads of the hard disc, before you switch off. **A**

Modula-2 Compiler

A complete interactive environment for developing modula-2 software

Complete debugging environment
Maths and system libraries
Full screen editor
Produces optimized stand alone code
Independent compilation for modules
Full Arthur and WIMP support
Numerous example programs with documentation
Supports hardware floating point processors

Price £89.95

RISCFORTH

Riscforth is a new, 32-bit implementation of the FORTH-83 standard, designed to take full advantage of the ARM Architecture.

Features include:

Multi-tasking	Optimizing Compiler
ARM Assembler	Full Screen Editor
File System Interface	Operating System Calls
Floating Point	Graphics Support
Hashed Dictionary Structure	Single-Step Debugger
Shadow Screen for Documentation	
Utilities for Block Manipulation, Dictionary and Vocabulary Display, Call Finding etc.	
Stand Alone Code Generation with no royalties payable	
Extensive Manual and many example programs	

Price £69.95

Blue Grey Software

28 Bobbitts Way, Wivenhoe, Essex CO7 9NJ

What's on the Menu? – WIMPs!

Adrian Look

There has been a large gap in this series but I shall continue now by explaining the menuing system employed by Arthur's Window Manager (AWM).

Menus are basically specialised windows which can contain text (or icons) selected by the user. The AWM will then indicate to the application which item has been selected.

To set up a menu hierarchy we must define the attributes of the menu and the items it contains. These include the menu's width, its title, the height of each item, and the vertical gap between these items. Each item can be ticked, shaded (i.e. it can't be selected by the user), writeable (used as a text input), have a dotted line after it, and/or point to a sub-menu.

'Wimp_CreateMenu' at &400D4

To define all these characteristics we have to use the 'Wimp_CreateMenu' routine. This is done by setting up the appropriate registers and a data block. The registers contain: the pointer to the data block (if this is -1 then the AWM will shut down all menus) and the position of the top-left hand corner of the menu (denoted by x,y in the example below). The data block should then be set-up as follows:

```
$block : title (12 bytes)
block?12: menu title foreground
block?13: menu title background
block?14: menu work area foreground
block?15: menu work area background
block!16: menu width
block!20: height of the following menu items
block!24: vertical gap between items (and at top
         and bottom of menu)
```

followed by the menu items (24 bytes each)

```
0-3 : flags    &01 item is ticked
        &02 dotted line displayed after item
        &04 item is writeable for text entry
        &80 indicates this is the last item in the menu.
4-7 : pointer to sub-menu data block (-1 if none)
8-11 : flags : &00000001 item is text
        &00000008 text is horizontally centred
        &00000010 text is vertically centred
        &00000020 item has a filled background
        &00000200 text is right justified
```

&40000000 item is shaded

12-23 : menu text (12 bytes)

Here is an example of how to use SYS "Wimp_CreateMenu":

```
DIM menu_block% &2000
:
DEFPROCcreate_menu
PROCset_up_block
SYS "Wimp_CreateMenu",,
                                menu_block%,x,y
ENDPROC
:
DEFPROCset_up_block
REM set up the data
REM at location menu_block%
REM as shown above
ENDPROC
```

Having called SYS "Wimp_CreateMenu", the AWM will attempt to open (display) the menu created. However, this is done via the 'Wimp_Poll' routine, and as menus are just specialised windows they will also require the 'window open' and 'window close' routines (see Archive 1.2 page 19).

'Wimp_Poll' – Menu_Select (9)

This brings us onto the Wimp_Poll condition (see vol.1 no.3 pg.26) that is unique to menus. When the user selects a menu item, the AWM will inform the application via 'Wimp_Poll' condition number nine. This condition is returned with the data block set-up as follows:

```
block!0 item number in first menu level (starting
at zero)
block!4 item number in second menu level
etc
(terminated by -1)
```

By using a series of CASE statements you can easily allow the application to 'decode' which item has been selected (see line 1610 in the listing below).

'Wimp_DecodeMenu' at &400D5

If you do not like handling all these numbers, then you can get the AWM to process them into a string. This is done by using the 'Wimp_DecodeMenu' routine. However, this routine requires you to keep the menu data structure in memory.

On entry:

- R1 = pointer to menu data structure
- R2 = pointer to list of menu selections
- R3 = pointer to buffer to contain the answer

On exit:

- R3 = pointer to string (being the menu items separated by a '.')

```
REM * for example *
DIM menu_structure% &2000
DIM block% &100
DIM string% &100
:
DEFPROCdecode_menu
SYS "Wimp_DeCodeMenu",,menu_
    structure%,block%,string%
PRINT $string%
ENDPROC
```

A few tips on how to use menus:

- For those of you who want to use 'writeable' menus (having set flag &40 – bytes 0-3) you will find that your menu item can be written to, but processing this is difficult and cumbersome. The easy way to do this is to direct the input to a data block and let the AWM take care of it. This is done by setting the flag &100 (bytes 4-7), and then where the 12 bytes of menu text would be, you should put:

```
text!0=instring%
text!4=-1
text!8=length
```

where a DIM in_string% length has been performed. You can access your input by means of \$in_string% (see the listing, lines 1160-1180).

- If you want to use different menus for different windows, use the Wimp_Poll condition 6 (Mouse_Button_Clicked) to test whether the menu button has been pressed; and by keeping track of Wimp_Poll conditions 4 and 5 (pointer entering/leaving window) you can tell which window the pointer is presently over. Thus you can call the appropriate create_menu and decode_menu procedures.

- I strongly urge you to write a 'short-hand' for creating menus rather than poking directly to the menu data block. This will save a lot of effort.

In conclusion, I recommend that you at least study the listing below, if not type it in and experiment! Next month, I shall look at icons. Happy menuing, and if you have any queries, write in and Paul will pass them on to me.

```
10 REM >$ .Wimp.Menu
20
30 REM *****
40 REM * Menus in the WIMP System *
50 REM * By Adrian Philip Look *
60 REM * 13th July 1988 *
70 REM *****
80
90 ON ERROR OSCLI("FX4,0"):MODE12
:PRINT REPORT$;" at line ";ERL:END
100
110 MODE12
120 *PRINT $.Wimp.WimpFont : REM Not
    imperative but they do
130 *PRINT $.Wimp.!Palette : REM make
    the WIMP much more presentable.
140 PROCvariables
150 version=FNwimp_initialise
160
170 PROCmain_menu : REM create menu
    hierarchy
180
190 REM main polling loop
200 SYS "Wimp_ForceRedraw",-1,0,0,
    1280,1024
210 *POINTER
220 REPEAT
230 SYS"Wimp_Poll",,block% TO reason
240 PROCwimp_poll(reason)
250 UNTIL FALSE
260 END
270
280 DEFPROCvariables
290 DIM block% &100 : REM general data
    block
300 DIM menu% &2000 : REM menu structure
    data block
310 DIM text% 12 : REM a data block
    for holding my text entry
320
330 DIM text$(50),setting(50),
    flags(50),data(50)
340 DIM submenu(50),ref(50),
    menupos(20)
350 menu_title_fg=13 : REM colours
360 menu_title_bg=12
370 menu_work_fg=7
380 menu_work_bg=0
390
400 tick=&01 : REM flag values
410 dotted=&02
420 writeable=&4
430 last=&80
```

WIMP Menus

```

440 default=&10021
450 shaded=&400000
460
470 PROCmenu_variables
480 ENDPROC
490
500 DEFPROCmenu_variables
510 first=TRUE
520 second=FALSE
530 sel=TRUE
540 $text%="!default"
550 ENDPROC
560
570 DEFPROCmain_menu : REM create
    'main' menu using 'short-hand'
580 no=0:MOUSE x,y,b
590 REM menu0 (main menu)
600 PROCdefine_menu_text("first",no,
    0,first,0,0,default,-1,0):no+=1
610 PROCdefine_menu_text("second",no
    ,0,second,-1,0,default,-1,0):no+=1
620 PROCdefine_menu_text("not here",
    no,0,0,0,-1,default,-1,0):no+=1
630 PROCdefine_menu_text("action",no
    ,0,0,-1,0,default,1,0):no+=1
640 PROCdefine_menu_text("just",no,0
    ,0,0,NOT sel,default,-1,0):no+=1
650 PROCdefine_menu_text("some",no,0
    ,0,0,NOT sel,default,-1,0):no+=1
660 PROCdefine_menu_text("text",no,0
    ,0,0,NOT sel,default,2,-1):no+=1
670 REM menu1
680 PROCdefine_menu_text("menu1",no,
    0,0,0,0,0,-1,-1):no+=1
690 PROCdefine_menu_text("deselect",
    no,0,NOT sel,0,NOT sel,
    default,-1,0):no+=1
700 PROCdefine_menu_text("select",no,
    0,sel,-1,sel,default,-1,0):no+=1
710 PROCdefine_menu_text("leave",no,
    0,0,0,0,default,-1,-1):no+=1
720 REM menu2
730 PROCdefine_menu_text("text",no,0
    ,0,0,0,0,-1,-1):no+=1
740 PROCdefine_menu_text("text%(12)",
    ,no,-1,0,0,0,default,-1,-1):no+=1
750 REM create menu structure
760 PROCcreate_menu("adrian",no,x,y)
770 ENDPROC
780
790 REM decoding the short hand
800 DEFPROCdefine_menu_text(text$,n,
    w,t,d,s,flags,sm,l)
810 text$(n)=text$
820 setting(n)=0
830 IF w THEN setting(n)=setting(n)
    OR writeable:flags=flags OR &100
840 IF t THEN setting(n)=setting(n)
    OR tick
850 IF d THEN setting(n)=setting(n)
    OR dotted
860 IF l THEN setting(n)=setting(n)
    OR last
870 IF s THEN flags=flags OR shaded
880 flags(n)=flags
890 submenu(n)=sm
900 ENDPROC
910
920 REM creating the menu structure
    data block
930 DEFPROCcreate_menu(title$,n,x,y)
940 mb%=menu%
950 menus=0:q=0
960 REPEAT
970 menupos(menus)=mb%:menus+=1
980 $mb%=title$
990 mb%?12=menu_title_fg
1000 mb%?13=menu_title_bg
1010 mb%?14=menu_work_fg
1020 mb%?15=menu_work_bg
1030 mb%!16=224 : REM width of menu
    items
1040 mb%!20=40 : REM height of
    menu items
1050 mb%!24=10 : REM vertical gap
    between items (also top
    & bottom)
1060
1070 itemptr=mb%+28
1080 REPEAT
1090 itemptr!0=setting(q)
1100 ref(q)=itemptr+4
1110 itemptr!8=flags(q) OR
    menu_work_bg<<28 OR
    menu_work_fg<<24
1120 IF (flags(q) AND &100)=0 THEN
    $(itemptr+12)=text$(q)
1130 ELSE
1140 pos=INSTR(text$(q),"(")
1150 itemptr!12=EVAL(LEFT$(text$
    (q),pos-1))
1160
1170 itemptr!16=-1
1180 itemptr!20=EVAL(RIGHT$(text$
    (q),LEN(text$(q))-pos))
1190 ENDIF
1200 itemptr+=24:q+=1

```


Oramics – Graphics to Sound

Daphne Oram

How exciting it is that home computer music is steadily moving towards "Oramics". Ian Nicholls' future development suggestions (on page 43 of his article "Sound Synthesis", Archive No.10, July '88) point clearly towards graphical control of all musical parameters.

Oramics is just that

The new Grove's Dictionary of Music shows a photograph of how I managed this graphical control in the early sixties – freehand input, a mixture of analogue (C.R.T scan) and digital read out. Sounds from this equipment were heard in London Concert Halls and a Ballet using them toured from Spain to South America via the Far East. In the seventies Canadian TV gave a performance of a new piano 'concerto', the 'orchestra' being Oramics.

Now on Archimedes

In the early 1980's I had hardware made to transfer the idea to Apple II+. It worked, but was too slow for intricate detail. However now, Archimedes, without extra hardware, should give us all we need in speed and memory. I have it already working with graphic input, but there is much more work to do on the various para-meters to get a really musical interface (for the 'serious' composer). How exciting to have 4096 pitches to the octave, but how tricky to recognise and control them!

Archimedes-Oramics was used by 1st year B.A. and B.Ed. students at a course for advanced musicians in June 1988. "Only took me half an hour to get the hang of it", said one girl student whose

only previous experience was a little word-processing. She was 'hooked'!

Intricate nuances

In the past, I have held English and American patents for the original Oramics. I hope, now that it is emerging, it will be used musically and with great respect for those truly wonderful instruments – the human ears. We can explore the most intricate nuances that, probably, the ear/brain can detect, and find out which are meaningful. All the time the VDU can display the graphic patterns, which produce the sounds, allowing subtle editing for greater refinement or modifying the content.

A Challenge

Oramics should give you weeks, months of fascinating enjoyment. Ever since 1944 when I first set my eyes on an oscilloscope and immediately wanted to reverse its function (graph in, sound out instead of sound in, graph out), I have been enthralled. It is most fulfilling. Best of luck to you all – do work out other editions of Oramics and let me know of your successes (via the editor), but I am an O.A.P. kneedeep in ARM assembly, (*Pun intended? Ed.*) so please forgive short replies!

Bibliography

Electronic Music Production A. Douglas USA 1982
Electronic & Computer Music P. Manning O.U.P 1985
"An Individual Note of Music, Sound & Electronics" D. Oram published Galliard, London 1972 also Galaxy Music Corporation, New York
Grove's Dictionary of Music **A**

```
1210 UNTIL (setting(q-1) AND last)>0
1220 mb%=itemptr
1230 IF q<>n THEN title$=text$(q):
      q+=1
1240 UNTIL q=n
1250
1260 FOR q=0 TO n-1
1270 IF submenu(q)>0 AND submenu(q)
      <menus THEN
1280 !ref(q)=menupos(submenu(q))
1290 ELSE
1300 !ref(q)=-1
```

```
1310 ENDIF
1320 NEXT q
1330 SYS "Wimp_CreateMenu",,menu%,x,y
1340 ENDPROC
1350
1360 REM servicing the Wimp_Poll
      routine
1370 DEFPROCwimp_poll(reason)
1380 CASE reason OF
1390 WHEN 2 : PROCopen_window
      (!block%)
```

Continued on page 51

ARM Assembler Floating Point Instructions

Ken Robbins

Introduction

A curious and widely regretted omission from the facilities of the ARM BASIC Assembler is any support for the floating point co-processor instructions. Even though the co-processor is not yet announced, its facilities are available by means of the Floating Point Emulator module and the absence of any assembly language support makes this very difficult to use. Fortunately, the ARM BASIC Assembler provides a kind of macro facility similar to that found in many dedicated assemblers and this is the solution to the problem.

A fully-fledged macro assembler allows the programmer to create macro instructions that are in every way indistinguishable from actual machine instructions but the ARM BASIC Assembler cannot do this; instead it allows the programmer to code a BASIC function reference whenever an instruction mnemonic would appear. This BASIC function may recursively enter assembly mode to create the instruction or data sequences required by the "macro". Peter Cockerell in his "ARM Assembly Language Programming" gives an excellent overview of the facilities for those wishing further information. The rest of this article will describe a library of BASIC functions which can assemble all of the floating point instructions documented in the Programmer's Reference Manual and also assemble a useful subset of floating point constants.

The "Macros"

There is one function for assembling all of the floating point instructions, `FNflop`, and it takes one parameter – a string containing the free-form text of a single floating point instruction, for example:

```
FNflop ("ADFSZ F1,F2,#0.5")
```

The function generates a full-word integer equal to the encoded floating point instruction and assembles it directly at the current location counter position. All operand fields are evaluated as expressions so "equated symbols" may be used to name registers and express literal values. `FNflop` recognises F0-F7 and R0-R15 automatically as register names. There are two functions for assembling floating point constants: these are "eqfS" for

single length numbers and "eqfD" for double length numbers. Each takes one parameter, an expression which will evaluate as a real number, e.g.

```
FNeqfS (3.14159)
```

Both functions work by mapping the ARMB floating point format onto the IEEE format (see Cockerell for details of the number formats). Support for extended length and packed decimal constants has been omitted because ARMB floating point numbers have insufficient precision or dynamic range for the extra formats to be necessary for data input and all internal computation is performed at full working precision anyway.

Using the Macros

Using the function library is straightforward. On your program development disk build a directory "\$MACLIB" and key or copy the "FloatOps" library file into it. In the BASIC program that you use to perform your assembly, add the statement:

```
LIBRARY "$MACLIB.FloatOps"
```

so it will be executed before the assembly begins; the sample program `ASSEMENV` illustrates the point. Then, in your assembler source code, whenever you would write a floating point instruction, write it enclosed in quotes as a parameter of function `FNflop`; whenever you would write a short floating point constant, write its value as a parameter of function `FNeqfS`; for a long floating point constant do the same with function `FNeqfD`.

The functions require the assembler pass number (which encodes the various assembler options such as "errors on/off") to be presented in the global integer variable `Pass%`; all assembler options are honoured by the functions. Assemble your object code by `RUN`ning the BASIC driver as usual. If you have enabled the listing option, the assembled floating point instructions and constants will appear in the usual way. The listing line will contain a row of ++'s to indicate output generated by the function.

Labels may be applied to both constants and instructions, and may be referred to by other instructions. An unfortunate artefact of the ARMB Assembler is that, when a function is encountered, it is listed after it has been processed and with the

current value of the location counter (i.e. as updated by the function's processing). This does not matter much unless a label is placed on the function; an incautious glance at the assembly listing then suggests the wrong value for the label. The label is actually processed before the function so its value is right – it just looks wrong. You can avoid confusion by putting the label on a separate line or by following it with a colon; either way, the ARMB Assembler will then list the label (with its correct value) before the function is processed.

A further difficulty with labels can arise if the proper four-byte boundary alignment for instructions is not observed. The FNflop, FNequfS and FNequfD functions all assemble their output at a four-byte boundary, inserting slack bytes if necessary to achieve this. However, any label on the function has already been processed and will have an incorrect value if boundary alignment takes place. This effect is not peculiar to the processing of functions; exactly the same thing happens when the ARMB Assembler processes a native instruction on a misaligned boundary.

Running with the Emulator

When your object program is complete, it can be executed in the usual way. Do not forget to load the Floating Point Emulator first or your program won't work. The emulator is in the "Modules" directory of the Welcome disk; insert the working copy of your Welcome disk and enter the command:

```
*RMLoad $.Modules.FPEmulator
```

You will receive an acknowledgement message when the emulator has initialised itself. You only need to do this once in each session of work.

Observations and Caveats

The method of testing used for this function was primarily to assemble code using the function, load the object code into storage, disassemble it with the debugger, and check that the debugger showed the same instructions as were originally assembled. All op-codes and all operand modes, including encoded literals, have been checked out although not in exhaustive combinations. The function makes a fair stab at syntax checking and appears to handle all correct input. However, it has not been systematically tested for the rejection of incorrect input although it does detect many of the common errors.

The floating point instructions generated by FNflop are as defined in the PRM, with the extra information on the op-code values for the CMF and CNF instructions that was determined by John Smith (Archive 1 No 7, page 32.)

During the development of the function, a number of inconsistencies between Cockerell, the PRM and the DEBUG program were found; in such cases, (hopefully) intelligent guesses were made to resolve them and these are documented here:

- The PRM suggests that the FLT instruction will process a literal value as its source argument but this is not supported by Cockerell and the DEBUG program. Since processing an encoded literal would duplicate a function already provided by the MVF instruction, PRM is probably wrong and FNflop rejects a literal value presented as a source argument.
- The PRM suggests that the syntax of the FLT instruction is: "FLT<p> <FPreg>,<ARMreg>" but the debugger does not support this, producing output of the form "FLT<p> <ARMreg>,<FPreg>". Cockerell agrees with the debugger in his general discussion of the co-processor data transfer instructions (p.182), but agrees with the PRM in his specific description of the register transfer instructions (p.196). The FNflop function follows the PRM on the ground that the destination register is conventionally specified first in all other register-register instructions.
- Following the PRM convention for register transfer instructions, that Fn is bits 16-18, Fm is bits 0-2, and Rd is bits 12-15, the PRM suggests that the syntax for the FIX instruction should be: "FIX<p> Rd,Fn", but the debugger does not support this, producing output consistent with the syntax: "FIX<p> Rd,Fm". The FNflop function follows the debugger on the ground that elsewhere, the convention is to use the Fm field as a source field. The correct operation of a small test program has confirmed this view.
- For the instructions WFS, RFS, WFC, RFC, the debugger appends an "S" modifier to the disassembled op-code but this is not documented anywhere. Experimentation with some related codes suggests that it is supposed to represent a precision. This is absurd; there is no arithmetic precision associated with the FPU status and

control registers. The debugger appears to be in error and the FNflop function does not support a precision modifier for these four instructions.

Concluding Remarks

Access to the facilities of the Floating Point

Emulator should increase the scope for Scientific programming in Assembly Language on the Archimedes. The author would welcome any feedback (via the columns of "Archive") on errors in the functions, particularly on correct resolutions of the inconsistencies described above.

```

10 REM > ASSEMENV
20 NAME$="ASSEMENV"
30 VDU 2                                :REM Turn on printer for assembly listing.
40 LIBRARY "$.MACLIB.FloatOps"         :REM Load floating point macro functions.
50 DIM Code% 16384                     :REM Default to 16K of object code.
60 Listing%=1                          :REM Assembler listing option.
70 Errors%=2                           :REM Assembler errors option.
80 Offset%=4                           :REM Assembler offset assembly option.
120 REM *****
130 REM *           Here we perform a two-pass offset assembly.           *
140 REM *****
150 FOR Pass%=Offset% TO Offset%+Errors%+Listing% STEP Errors%+Listing%
160 O%=Code%                            :REM Initialise location counter.
170 P%=&8000                            :REM Load point in application work-space.
180 [
190     OPT Pass%                        ; Set assembly options for pass
200 .FPTests MOV R9,#10                 ; Pick up a constant.
210     FNflop ("FLTD F0,R9")           ; Convert it to float.
220     FNflop ("STFD F0,Float2")       ; Save it for inspection.
230 .TakeRoot:FNflop ("SQTD F1,F0")     ; Take the square root.
240     FNflop ("STFP F1,Packed1")      ; Save it for inspection.
241     FNflop ("ADFSZ F2,F1,#0.5")     ; Add a literal value.
242     FNflop ("STFE F2,Float5")       ; Save it for inspection.
250     SWI "OS_Exit"                   ; Exit so we can have a look.
260 .Float1 : FNequfD (10)
270 .Float2 : FNequfD (0)
280 .Float2a: FNequfS (0)
290 .Float3 : FNequfS (3.1415927)
300 .Float4 : FNequfD (3.1415927)
301 .Float5  EQU D 0:EQU D 0:EQU D 0    ; Extended float is 3 words.
310 .Packed1 EQU D 0:EQU D 0:EQU D 0    ; Packed decimal is 3 words.
340     EQU D 0
350     EQU D 0
360 ]
370 NEXT Pass%
380 VDU 3                                :REM Turn the printer off again.
390 REM *****
400 REM * Here we save the object code as an absolute application program*
410 REM * (type FF8) in the LOAD sub-directory of the current directory.*
420 REM *****
430 OSCLI "SAVE LOAD."+NAME$+" "+STR$(Code%)+ " "+STR$(O%)
440 OSCLI "STAMP LOAD."+NAME$
450 OSCLI "SETTYPE LOAD."+NAME$+" &FF8"
460 END

```

(Unfortunately, there just is not room for the main listing – even after stripping out the copious REMs, it occupies 4+ pages. If you want a copy of the commented listing, send a large S.A.E. Ed.) **A**

Now available

£105 from Archive

PipeDream

PipeDream is now available on the Acorn Archimedes. It provides comprehensive word processing, spreadsheet and database facilities integrated in a way only dreamed of before.

With other integrated packages, you have to divide your work into artificial sections, such as text, numbers and calculation, and database.

With PipeDream, you compose your work in the order you want to print it, with text and numbers all together in one document. Incorporate calculations directly into paragraphs of text and formatted paragraphs directly into spreadsheets.

PipeDream is ideal for all tasks involving words and numbers. From writing thank-you letters to encyclopaediae, invoices to cash flow forecasts, stamp-collection records to inventory management, film scripts to mail-shots.

PipeDream is 100% file and keystroke compatible with Z88 PipeDream and PipeDream on the IBM PC. It is also compatible with View Professional on the BBC microcomputer. You can create documents on any of these computers, transfer the files to any other and continue working, with no loss of information. No other software enables you to share your files with all these computers.

PipeDream for the Acorn Archimedes costs £99 + VAT.

It is not possible to detail all of PipeDream's facilities here. For full details or to order PipeDream call us on 0954 211472 or write to us at Colton Software, Broadway House, 149-151 St Neots Road, Cambridge CB3 7QJ.

PipeDream - power at your fingertips.

Unpacking Packed B.C.D.

Gerald Fitton

With Acorn's Floating Point Emulator you can do sums to about 19 digit precision. Probably the best interface between the FPE and BASIC is Packed Binary Coded Decimal. The program "Pi" shows how, having obtained the result of a floating point calculation in packed BCD, it can be unpacked into a BASIC string. The routine which does this is FNFPAss at line 640 of the listing below. Before trying to RUN it, you need to *RMLOAD the FPE. (Version 2.50 of the Floating Point Emulator is included on the monthly program disc.)

The program has a few other interesting features.

Lines 130 to 160 provide a legal way of ensuring that the code will always be assembled at &10000, thus avoiding page &FF00. Page &FF00 can sometimes cause problems since Acorn used it on their 8-bit machines for Input/Output and they have retained upgrade compatibility.

You can calculate π by finding the polar angle (measured anti-clockwise from the positive x-axis) of the point with co-ordinates $x=-1$, $y=0$. This angle, in degrees is 180, which converts to exactly π radians. Mathematicians familiar with complex numbers will recognise this angle as $\text{Arg}(-1,0)$.

The assembly language program for finding π is contained in lines 460 to 590. It contains FP opcodes assembled with the EQU directive because BASIC does not support FP mnemonics.

This unpacking routine forms part of the Floating Point Assembler Software Package which can be purchased from Abacus Training and the program has been adapted from one of the demonstration programs supplied. Their FP package is designed to let you assemble FP opcodes from FP mnemonics.

```
10 REM > Pi
20 REM Author : G L Fitton
30 REM Copyright : ABACUS TRAINING
40 REM Version 1.00: 19th July 1988
50 :
60 REM Finds the value of pi to
  about 19 significant figures.
70 REM Uses Floating Point Emulator
  Version 2.50
```

```
80 REM with double extended
  precision and returns the
90 REM answer as a 19 digit number
100 MODE 0
110 :
120 REM Store the object code at
  &10000.
130 size%=&100
140 DIM dummy1% 0
150 DIM dummy2% (&10000-&28-dummy1%)
160 DIM code% size%
170 :
180 end%=FN_Pi(code%):REM Assemble
  code at &10000
190 CALL code% :REM Find Pi as pbcd
200 PRINT
210 REM Show code as disassembled by
  the Debugger.
220 OSCLI("MEMORYI "+STR$(code%)+
  " +2C")
230 :
240 REM Unpack pbcd version of Pi
  and print answer as 19 digits.
250 PRINT "'The value of pi is ";
  FN_FpAss_Unpack(pbcd,19)
260 END
270 :
280 DEF FN_Pi(code%)
290 LOCAL sp,link
300 LOCAL pass%
310 :
320 pbcd=code% : REM Avoid an error
  on the first pass.
330 :
340 REM ARM registers assigned to
  variables.
350 REM Address of pbcd version of
  pi stored in R0.
360 sp= 13 :REM Use BASIC stack.
370 link = 14 :REM Link back to BASIC
380 REM Floating Point registers
  assigned to variables.
390 REM Polar angle stored in F0.
400 REM x co-ordinate stored in F1.
410 REM y co-ordinate stored in F2.
420 REM Value of pi stored in F3.
430 :
```



```

440 FOR pass%=0 TO 3 STEP 3
450   P%=code%.
460   [OPT pass%
470   STMFD (sp)!,{R0-R12,link}
           ; Store ARM registers
480   EQUd &EE08A108   ; F2 = 0
490   EQUd &EE189109   ; F1 = -1
500   EQUd &EEC90102   ; POL(-1.0)
510   EQUd &EE08B100   ; F3 = F0
520   ADR R0,pbcd      ; Address for
                       answer
530   EQUd &ECC0B100   ; pbcd = F3
540   LDMFD (sp)!,{R0-R12,PC} ; Back
                       to BASIC
550   .pbcd
560   EQUd 0           ; Answer packed here
570   EQUd 0
580   EQUd 0
590   ]
600 NEXT pass%
610 :
620 =P%
630 :
640 DEF FN_FpAss_Unpack(address%,
                       digits%)
650 REM Unpacks a BCD number which
     is stored at address%.
660 REM If the number can not be
     expressed in <digits> digits
670 REM then it is returned in
     scientific notation.
680 :
690 LOCAL bcd$,exponent$,mantissa$,
     mantissal$
700 LOCAL scientific$,sign$,unpacked$
     ,word%
710 :
720 REM Collect the pbcd number from
     address%.
730 FOR word%=address% TO address%+8
     STEP 4
740   bcd$=bcd$+RIGHT$("00000000"+
     STR$(~(!word%)),8)
750 NEXT word%
760 :
770 REM Split into mantissa, exponent
     and sign.
780 mantissa$=RIGHT$(bcd$,19)
790 mantissal$=LEFT$(mantissa$,1)+
     "."+RIGHT$(mantissa$,18)
800 exponent$=MID$(bcd$,2,4)

810 sign$=LEFT$(bcd$,1)
820 :
830 REM Concatenate into scientific
     notation.
840 IF INSTR("8C",sign$) THEN
     scientific$="-"+mantissal$ ELSE
     scientific$="."+mantissal$
850 IF INSTR("4C",sign$) THEN
     scientific$+=" E-"+exponent$ ELSE
     scientific$+=" E"+exponent$
860 :
870 REM If the number has fewer
     than <digits> digits
880 REM then use standard rather
     than scientific notation.
890 IF VAL(exponent$)<digits% THEN
900   mantissa$=mantissa$+STRING$
     (digits%,"0")
910   IF INSTR("4C",sign$) THEN
920     unpacked$="0."+STRING$(VAL(
     exponent$)-1,"0")+mantissa$
930   ELSE
940     unpacked$=LEFT$(mantissa$,
     VAL(exponent$)+1)+"."+MID$
     (mantissa$,VAL(exponent$)+2)
950   ENDIF
960 :
970 REM Strip trailing zeros.
980 REPEAT
990   IF RIGHT$(unpacked$,1)="0"
     THEN unpacked$=LEFT$
     (unpacked$,LEN(unpacked$)-1)
1000 UNTIL RIGHT$(unpacked$,1)<>"0"
1010 :
1020 REM Strip trailing decimal pt
1030 IF RIGHT$(unpacked$,1)=". "
     THEN unpacked$=LEFT$(unpacked$,
     LEN(unpacked$)-1)
1040 :
1050 REM Affix the correct sign.
1060 IF INSTR("8C",sign$) THEN
1070   unpacked$="-"+unpacked$
1080   ELSE
1090   unpacked$="."+unpacked$
1100 ENDIF
1110 ELSE
1120   unpacked$=scientific$
1130 ENDIF
1140 =unpacked$ A

```

Alphabetic Alternatives

Brian Cowan & Ian Smith

The article in the June issue on using View mentioned the "top bit set" characters. We explained what they were, why they were useful and the article finished by saying that printing this extra set of characters was not simple; this was to be discussed in a future article. Although this present article is ostensibly about using View, the ideas in it are more generally applicable. Also, some of the techniques could well, on the authors' own admission, be improved upon, so please send in your own ideas.

Alphabets

Archimedes provides eight different screen alphabets. The "top bit un-set" characters of these correspond essentially to the conventional ASCII standards. The extra characters provide most of the "extended character set" definitions; there are four Latin sets, a Greek set and, for compatibility with earlier BBC machines, the BFONT and the Teletext text and graphics character sets. This represents quite a choice of screen displays for the user. Once the screen alphabet is chosen, using *alphabet for example, the extended characters are produced, as explained last month, by pressing the <ctrl>, <shift> and <alt> keys. (Hold down <ctrl> and <shift> and then press <alt>.)

Printing

So what is printed on the printer? The printer does not know which alphabet is selected. All it sees is a stream of eight bit codes. The usual "top bit un-set" codes represent the usual well defined ASCII characters. The others depend on the printer. Some printers provide a choice of character sets. Unfortunately none corresponds to any of the alphabets of the Archimedes, e.g. the Epson LX 80, 86 and 800 models provide an italic set and the IBM standard graphics set for the extra characters.

Ensuring WYSIWYG

To get the same display on screen and paper, a modest prerequisite for painless wordprocessing, something must be done! Either the printer's characters must be redefined or those of the

computer display must be changed. The former is difficult and varies from printer to printer. Luckily the latter is possible and not too difficult. Since most printers do provide the IBM character set it is surprising that the Archimedes does not. The central problem then is to provide redefinition of the Archimedes screen character set. In other words we must provide it with a new alphabet – obviously the IBM character set will have many uses other than simply with View wordprocessing

Creating a View Disc

Remember that the spirit of June's article was to produce a View disc that could be booted up. First let's look at what is on the new disc and then we will explain what is going on. The root directory \$ of the disc contains the !BOOT file together with directories called LIBRARY and TEXT. The !BOOT file consists of the following: (The line numbers are for reference only; they are not included in the file).

- 1 *BASIC
- 2 *MOUNT
- 3 *DIR LIBRARY
- 4 CHAIN "KEYS"

The business part of the proceedings is contained in the LIBRARY directory. Line 1 ensures the system is in BASIC since at line 4 the BASIC program KEYS is being CHAINED.

Redefining the Screen Characters

Each screen character is represented by an 8 by 8 matrix of dots. It can thus be specified by eight bytes. From BASIC, the screen characters may be redefined using VDU 23. For each character, eight numbers between zero and 255 are supplied which define the dot pattern for each of the eight rows. The program on this month's magazine disc called CHDEFS provides an early version of the character redefinitions done in this way. As an alternative, the Welcome disc on the BBC Master 128 provides a BASIC program called CHARDES which allows you to redefine screen characters in an easy and rapid manner. It has the advantage that you can see what the character looks like as you build it up dot

by dot. A disadvantage is that the machine code file so produced is 6502 code so it must be run under the emulator. Our KEYS program does the redefining of the screen characters to the IBM set. It consists of lines of VDU 23 statements such as

VDU 23,180,24,24,24,248,24,24,24,24

This line is redefining the character whose ASCII code is 180. In binary, the last eight numbers give the row by row pixel pattern for the character. The KEYS program ends with the line *EXEC !VIEW. The !VIEW file contains the following:

```

1 QUIT
2 RMRUN 65Arthur
3 *LOAD VIEW 8000
4 *GO 8000
5 *ECHO <26>
6 *FX 113 2
7 *ECHO <17><0>
8 *FX 112 2
9 *SCREENLOAD SCREEN
10 *FX 112 0
11 *FX 113 0
12 *KEY 11 "MODE 0|M *FX 220 32|M
    *FX 113 2|M"
13 *KEY 12 "*FX 113 0|M
    *FX 220 27|M MODE 0|M"
14 SETUP F I J
15 LOAD BLANK
16 *MOUNT 0
17 *DIR TEXT
18 *FX 202,16
19 *FX 118
20 NAME No_name
21 MODE 0
22 SCREEN $.LIBRARY.MSG

```

Again the line numbers are for reference only. We see that lines 1 to 4 enter the emulator and put the system into View in the familiar way. Line numbers 5 onwards are concerned with the provision of a Help Screen for the extra characters.

Making a Help screen

The final refinement is a screen to show what each key produces following depression of the <ctrl>, <shift> and <alt> keys. There is a View file called KeyChDefs on the magazine disc which shows a keyboard with the various key functions. This was

inspired by John Niblett of D.A.Computers who produced a similar one for the Master Compact. However, for maximum convenience we want a readily available help screen that can be called up at the touch of a button. Function keys 12 and 13 are programmed to go into and out of the Help Screen. Use is made here of the screen bank switching facilities of the Archimedes using the *FX 112 and *FX 113 commands. *FX 112 n selects screen bank n to be addressed by the VDU driver and *FX 113 n selects the bank addressed by the display hardware. The default screen is number 0 and the Help Screen is number 2. Thus lines 5 to 9 of file !VIEW load in the help screen file called SCREEN into bank 2. The *ECHO command with the pointed brackets is a way of using the BASIC VDU statements outside of BASIC. Thus line 5 would be VDU 26 and it moves the screen cursor out of the way so that it does not corrupt the Help screen. Similarly, line 7 causes the screen messages to be typed in black! Once the Screen is loaded, lines 10 and 11 take you back to the main screen.

Function Keys

The function keys are programmed in lines 12 and 13. The liberal use of MODE 0 statements is used to clear the View Command Screen since otherwise it is possible for it to overflow into the Help Screen. The rest of the !VIEW file should be self explanatory. The SCREEN file was made from the KeyCh Defs file by *SCREENSAVEing it and then editing it using the sprite editor on the Welcome disc. The files !BOOT, !VIEW, KEYS, SCREEN, CHDEFS and KeyChDefs are all on this month's disc.

Conclusion

This is a rather cumbersome way of squeezing the last bit out of the View wordprocessor. Clearly the character redefinitions would work on the Master Compact and presumably on the Master 128 if you load in the newer version of View. There are probably some problems remaining concerning the implementation of the bank-switched Help Screen and things overflowing into it. Also, the Help Screen is entered from the Command screen of View and not the Text screen. Really this is more an exercise on learning about the Archimedes while we wait for First Word Plus. **A**

Compacting Screens

Adrian Look

While compiling the program and shareware discs I found that there really wasn't enough room for many screen shots, e.g. a Mandelbrot screen is 80k (mode 13) and on an 800k disc you can only fit 9 screens. So I wrote the screen compaction program.

In order to compact a screen we need only record the changes in colour. This is done by reading a byte from the screen and seeing how many times it is repeated. The two numbers are then stored, i.e. (quantity first byte, colour second). For example,

```
screen: 1 1 1 1 1 5 5 5 3 3 3 5 5
                                     (13 bytes)
```

```
compact: 5 1, 3 5, 3 3, 2 5
                                     (8 bytes)
```

The more frequent the changes of colour, the less efficient this method becomes. When every adjacent pixel on the screen is different, the 'compacted' screen end up TWICE the size of the original, for example:

```
screen: 1 2 3 4 5 6
compact: 1 1, 1 2, 1 3, 1 4, 1 5, 1 6
```

This is not good enough. In order to reduce this I split the 'quantity' byte into two sections. One bit is used as a flag and the other seven are the same as before. The 'flag' bit is used to indicate whether the colour values that follow are all different. If this bit is set, the quantity bits will indicate how many bytes are to be expected. Thus:

```
screen: 1 2 3 4 5 6 becomes
compact: &86(%10001010), 1, 2, 3, 4, 5, 6
and screen: 7 7 7 7 7 7 becomes
compact: &06(%0 0001010), 7
```

The overall effect is that, in the large majority of cases, the screen can be quite substantially reduced. An 80k Mandelbrot can be compacted to an average of about 30-40k. Easier screens can be reduced considerably more.

Another advantage of compacting the screens is that the data can be *LOADED into memory and then decompact - thereby drastically reducing the loading time. The 'compact load' is, on average, 3-4 times faster even than the Fast ScreenLoad

Module (for Mandelbrot screens), this means that a screen can be loaded in more like 2 seconds.

I admit that my algorithm is not the best and that the ARM coding is definitely not the most efficient. However, as it stands the program is very acceptable and further improvements, to either the algorithm or the coding, hardly seem worth the time and effort. Having said this, we would all like to hear from anyone who comes up with anything better.

```
10 REM >$.SCompact
20
30 REM *****
40 REM *Screen Compact/De-compact*
50 REM *   Adrian Philip Look   *
60 REM *       23rd June 1988     *
70 REM *****
80
90 DIM code% &1FF
100 DIM workspc% &2D062
110 DIM block% 16
120
130 MODE 0
140 INPUT "Source file>" s_file$
150 INPUT "Destination file>" d_file$
160 OFF
170
180 PROCassemble
190
200 TIME=0
210 OSCLI("FASTLOAD "+s_file$) : OFF
220 f=TIME
230
240 n=INT(FNsize/&400)
250
260 TIME=0
270 PROCsquash(d_file$)
280 s=TIME
290
300 TIME=0
310 PROCunsquash(d_file$)
320 l=TIME
330
340 PRINTTAB(1,1)"OldLoad>";f/100;"s"
350 PRINTTAB(1,2)"CompactLoad>";
                                     l/100;"s"
360 PRINTTAB(1,3)"Time factor>";
                                     INT(10*f/l)/10
```



```

370
380 c=INT((workspc%!*64+68)/400)
390
400 PRINTTAB(1,5)"Old: ";n;"k"
410 PRINTTAB(1,6)"Compact: ";c;"k"
420 PRINTTAB(1,7)"Compaction factor:
      ";INT(10*n/c)/10
430 PRINTTAB(1,9);
440 ON
450 END
460
470 DEFPROCsquash(file$)
480 PROCsave_palette
490 C%=workspc%
500 E%=FNscrnadr
510 F%=E%+FNsize
520 CALL squash
530 OSCLI("SAVE "+file$+" "+STR$~
      (workspc%)+""+STR$~(workspc%!*64
      +68))
540 ENDPROC
550
560 DEFPROCunsquash(file$)
570 OSCLI("LOAD "+file$+" "+STR$~
      (workspc%))
580 MODE workspc%?1 : OFF
590 !workspc%=0
600 C%=workspc%
610 E%=FNscrnadr
620 F%=E%+FNsize
630 CALL unsquash
640 ENDPROC
650
660 DEFPROCsave_palette
670 workspc%?0=22
680 workspc%?1=MODE
690 workspc%?2=0
700 workspc%?3=0
710 ws%=workspc%+4
720 FOR palette=0 TO 15
730   ws%?0=19
740   ws%?1=palette
750   SYS "OS_Word",11,ws%+1
760   ws%+=6
770 NEXT palette
780 ENDPROC
790
800 DEFFNscrnadr
810 block%!0=149
820 block%!4=-1
830 SYS "OS_ReadVduVariables",block%
      ,block%+8
840 =block%!8
850
860 DEFFNsize
870 SYS "OS_ReadModeVariable",MODE
      ,4 TO ,,x
880 x=4-x:x=8*(2^x)*10
890 SYS "OS_ReadModeVariable",MODE
      ,5 TO ,,y
900 y=3-y:y=y*256
910 SYS "OS_ReadModeVariable",MODE
      ,3 TO ,,c
920 CASE c OF
930   WHEN 1 : m=8
940   WHEN 3 : m=4
950   WHEN 15 : m=2
960   WHEN 63 : m=1
970 ENDCASE
980 =x*y/m
990
1000 DEFPROCassemble
1010 bytea=0
1020 byteb=1
1030 save=2
1040 add=3
1050 screen=4
1060 end=5
1070 init=6
1080
1090 FOR opt=0 TO 3 STEP 2
1100   P%=code%
1110   [OPT opt
1120     .squash
1130     STMTD R13!,{R0-R12,R14}
1140     MOV R7,#&FF
1150     ADD save,save,#&64
1160     MOV init,save
1170     ADD save,save,#4
1180     LDRB bytea,[screen],#1
1190     .start
1200     MOV add,#1
1210     .loop
1220     STRB R7,[screen,#-1]
1230     STRB bytea,[save,add]
1240     ADD add,add,#1
1250     MOV byteb,bytea
1260     LDRB bytea,[screen],#1
1270     CMP screen,end
1280     BEQ skip
1290     CMP add,#127
1300     BEQ skip
1310     CMP bytea,byteb
1320     BNE loop

```

```

1330 .skip
1340 CMP add,#2
1350 BEQ same
1360 CMP bytea,byteb
1370 SUBEQ add,add,#2
1380 CMP add,#127
1390 BNE not
1400 STRB bytea,[save,add]
1410 MOV byteb,bytea
1420 LDRB bytea,[screen],#1
1430 .not
1440 ADD add,add,#&80
1450 STRB add,[save]
1460 SUB add,add,#&7F
1470 ADD save,save,add
1480 CMP screen,end
1490 BEQ leave
1500 CMP bytea,byteb
1510 BNE start
1520 .same
1530 MOV add,#1
1540 .loop1
1550 ADD add,add,#1
1560 CMP add,#127
1570 BEQ skip1
1580 CMP screen,end
1590 BEQ finish
1600 LDRB bytea,[screen],#1
1610 CMP bytea,byteb
1620 BEQ loop1
1630 .skip1
1640 CMP add,#127
1650 ADDEQ screen,screen,#1
1660 STRB byteb,[save,#1]
1670 STRB add,[save]
1680 ADD save,save,#2
1690 B start
1700 .finish
1710 STRB add,[save]
1720 STRB byteb,[save,#1]
1730 ADD save,save,#2
1740 .leave
1750 SUB save,save,init
1760 SUB save,save,#4
1770 STR save,[init]
1780 LDMFD R13!,{R0-R12,PC}
1790
1800 .unsquash
1810 STMF R13!,{R0-R12,R14}
1820 MOV R0,save
1830 MOV R1,#&64
1840 SWI "OS_WriteN"

1850 ADD save,save,#&68
1860 .begin
1870 LDRB bytea,[save],#1
1880 CMP bytea,#&80
1890 BMI miss
1900 SUB bytea,bytea,#&80
1910 .loop4
1920 LDRB byteb,[save],#1
1930 STRB byteb,[screen],#1
1940 SUB bytea,bytea,#1
1950 CMP screen,end
1960 BEQ quit
1970 CMP bytea,#0
1980 BNE loop4
1990 B begin
2000 .miss
2010 LDRB byteb,[save],#1
2020 .loop5
2030 STRB byteb,[screen],#1
2040 SUB bytea,bytea,#1
2050 CMP screen,end
2060 BEQ quit
2070 CMP bytea,#0
2080 BNE loop5
2090 B begin
2100 .quit
2110 LDMFD R13!,{R0-R12,PC}
2120 ]
2130 NEXT opt
2140 ENDPROC A

```

mentioned – so this book could not (and indeed should not be expected to) replace the Acorn Programmer's Reference Guide.

This review was written around an early copy of the book, which contained a few minor mistakes. An errata sheet is being prepared and will be supplied with the book in due course.

Overall then, this is a comprehensive and wide ranging book which stands up well both as a tutorial on assembly language programming and as a guide to the programming environment and facilities provided by Arthur.

I recommend it particularly to people following my series on ARM programming as a reference book to explain in greater detail the things I have had to gloss over (as it is there are 8 parts in the series!).

'Archimedes Assembly Language' by Mike Ginns, published by Dabs Press. ISBN 1-870336-20-8, price £14.95 (£13.50 through Archive). **A**

New Assembly Language Book

Alan Glover

Until now there has been little choice about a reference work on programming the ARM (the processor inside the Archimedes). The standard reference work has been Peter Cockerell's book *ARM Assembly Language Programming*. — This book was published at about the same time that the Archimedes was announced. In order to get the book out that quickly it was written around the ARM Evaluation System (an early ARM CPU set up as a second processor to a BBC Computer). Consequently it contains no information specifically about the Archimedes or Arthur.

Mike Ginns' new book is different. It has been written for Archimedes users and contains a lot of extra information about using Arthur as well as the obligatory information about assembly language.

The first impression of the book is that it is almost double the thickness of the Cockerell book. In fact it runs to 368 pages, and contains 65 program examples. It is spiral bound, like other books from Dabs Press. A disc is available containing these programs and some other useful utilities — a total of 74 programs in all and costs £9.95 (£8.50 through Archive). Incidentally one nice touch is that the index includes these 65 programs, so a reader can rapidly find a routine wherever it is in the book.

The book begins with an overview of the ARM and proceeds on to the internal architecture of the processor. It then gives an introduction to using the assembler in BBC BASIC so that the user can assimilate and experiment with the example programs.

The next group of chapters works through the five instruction groups and all their possible syntactic arrangements. One thing which impressed me in this section is that each instruction is at the top of a page, so it is very easy to find a particular command. Similarly the conditional execution codes are presented two to a page, with one condition at the top of a page, and its logical inverse below it.

The actual explanations are lucid and remind me of the layout of the assembly language opcodes in the Advanced User Guide. Every instruction is detailed under the headings of syntax, operation, and flags

affected, followed by an explanation and an example. It is a very clear layout, making it easy to look toward the right bit of the page when seeking specific information rather than having to hunt through an entire section.

After all the ARM instructions have been covered (it does not cover the co-processor instructions) the book expands into the general field of using assembly language on the Archimedes.

A chapter explains how to use the Debugger module and other debugging techniques. It also contains (in bold print!) the first law of assembly programming — always save the source code before letting the machine code loose!

The following chapters consider interrupts, vectored routines, how to claim the vectors and SWI routines. The book only mentions SWIs within Arthur, presumably a demarcation point between this book and Dabs's forthcoming book about the Archimedes Operating System. A number of the calls are detailed, with example programs.

The next two chapters cover the WIMP environment and font facilities. The latter chapter contains one of the clearest explanations of anti-aliasing that I have seen. (It almost makes sense to me now!)

The remaining chapters show how to simulate BASIC structures using assembly language. The function of a BASIC command is explained and then an equivalent in assembly language is given. Apart from their intrinsic value, these routines form useful illustrations of real code — one of the problems of producing this kind of book (or writing a series about Assembly Language Programming for that matter!) is that the examples given are usually quite artificial, just meant to get across one point.

The book also has a number of appendices covering numeric representation, arithmetic in computers, logical operations, the structure of the 32 bit words which form ARM instructions, a listing of Arthur SWI calls, OSBYTE calls, OSWORD calls and VDU and Plot codes.

However, the tables of the SWI calls just contain the name and number, the parameters are not

Continued opposite

Assembly Language Programming – 2

Alan Glover

Last month we explained that each ARM instruction is one 32 bit word, with all the necessary parameters encoded in those 32 bits. To illustrate this we will spend most of this part considering a small program. Type this in:

```
10 DIM Code 200
20 P%=Code
30 [OPT 3
40 SWI "OS_WriteS"
50 EQU S "Press a key"
60 EQU B 0
70 ALIGN
80 SWI "OS_NewLine"
90 SWI "OS_ReadC"
100 SWI "OS_WriteS"
110 EQU S "You pressed : "
120 EQU B 0
130 ALIGN
140 SWI "OS_WriteC"
150 SWI "OS_NewLine"
160 MOV R15,R14
170 ]
180 CALL Code
```

When you are certain the program is entered correctly, RUN it. A prompt will be displayed asking for a key to be pressed. The program will then tell you what key you pressed and return to BASIC.

Let's take a line-by-line look at what is happening:

10 Defines some space for the program to reside. Note that this use of DIM reserves a number of bytes, rather than setting up an array. (DIM Code<space>200 as opposed to DIMCode(200) which would set up an array.)

20 Sets the Assembler's location counter (P%) so that the code is placed in the space reserved for it as it is assembled.

30 Sets the Assembler's option control. A value of 3 tells it to stop on errors which occur during the assembly and to list the code produced. The [is always used to mark the beginning of assembly language statements within a BASIC program.

40 This is the first instruction. SWI is a call to Arthur. "WriteS" is the name of the call to use.

This one writes all following bytes to the screen until a 0 is encountered.

50 The text string for SWI "WriteS". EQU S stores a string expression at the current value of P%

60 This is the 0 to terminate the text string. EQU B stores one byte at the current value of P%. Other similar commands are EQU W for two bytes and EQU D for four.

70 Every ARM instruction must begin at an address which is a multiple of 4. Since the text beforehand cannot be guaranteed to be an exact multiple of four bytes, the ALIGN instruction is used to increment P% until it is a multiple of 4.

80 This is another call to Arthur, this time to write a Carriage Return and Line Feed on the screen, i.e. start a new line, hence the name.

90 This call to Arthur waits until a character is pressed on the keyboard and returns the ASCII code in Register 0, usually called R0.

100 This is another call to the routine which writes text on the screen.

110 This is the text to be put on screen,

120 and this is the 0 to terminate,

130 and the ALIGN to get back to a multiple of four.

140 Another call to Arthur – It writes the character, whose ASCII number is in R0, to the screen.

150 This call starts a new line on the screen.

160 This instruction is used to leave the program and return to the caller of the routine. The ARM does not have a true GOSUB as such, but does have an instruction which sets up R14 as the address to return to after execution. By moving this value of R14 into the program counter, R15, our routine hands back execution to the caller.

170 The] symbol indicates the end of the assembly language section

180 This command calls the assembled code

When you typed RUN you would have a display something like :


```

>RUN
0000906C          OPT 3
0000906C EF000001 SWI "OS_WriteS"
00009070          EQU$ "Press a key"
0000907B 00       EQU$ 0
0000907C          ALIGN
0000907C EF000003 SWI "OS_NewLine"
00009080 EF000004 SWI "OS_ReadC"
00009084 EF000001 SWI "OS_WriteS"
00009088          EQU$ "You pressed: "
00009096 00       EQU$ 0
00009098          ALIGN
00009098 EF000000 SWI "OS_WriteC"
0000909C EF000003 SWI "OS_NewLine"
000090A0 E1A0F00E MOV R15,R14
Press a key <Press O>
You pressed : O

```

Reading across, the first number is the location at which the instruction is being stored, the current value of P%. This may well be different on your computer. The next column shows the actual assembly language instruction's value in machine code. The next column gives the assembly language statement. There is a final column, which is not used above, which can be used to hold comments about the program, just like REMs in BASIC. A comment can begin with either a / or ;, for example:

```
EQU$ 0 / End of WriteS text or,
```

```
ALIGN ; Next word boundary
```

You will notice that all the SWI instruction codes begin &EFxxxxxx. The xxxxxxxx is used to hold the number of function being requested. BASIC will attempt to find the number to use for a SWI name. That is why the case and spacing of the names within the quotes must be perfect.

Similarly, by looking at the code for MOV R15,R14, you will notice the F and E for registers 15 and 14 respectively. This underlines the point made earlier that, unlike some CPUs, the ARM has all the information needed to perform an instruction in one 32 bit word, rather than having to fetch one or more extra bytes to complete the instruction.

Also encoded in the instruction are four bits called the 'conditional field'. This is in fact the first digit, 'E' in all the instructions above.

This conditional field can be used to produce instructions which will only execute when a particular combination of the status flags is true. This allows instructions which need not be executed to be passed through and ignored, instead of having to skip around them.

At first that may not sound such a big deal, but consider again the 'pipelining' of the ARM. If the Program Counter is altered, the two subsequent instructions that it has fetched are now useless and have to be forgotten so that execution can begin elsewhere. At the new address, the execute logic will be inactive for two cycles and the decode logic for one cycle, while the new instructions are working through the pipeline.

For this reason conditional execution of instructions should be used in preference to jumping around instructions.

In the next part we will continue the theme of conditional execution with some examples. Remember that the best way to learn is to experiment – try altering the example program to do slightly different things, for example issue a VDU 7 with the 'Press a key' prompt to attract attention.

This could be done in one of two ways. The first is to put an EQU\$ 7 before the EQU\$ 0 in the first text string, e.g. at line 55.

The other way is to use another SWI call. This one is called "WriteI". It encodes not only the SWI number but the character to be written in the instruction too.

Unfortunately the expression evaluation of the SWI numbers make this a bit messy. The neatest way is to put in at line 15 WriteI=&100, which defines the number of the SWI, and use at line 85 SWI WriteI+7. The 7 represents the ASCII code to produce the VDU 7.

P.S. Last month I said that the ARM processor has a 32-bit address bus – in fact that should have read "data bus" – the data bus is 32 bits wide, but the address bus is actually 26 bits wide, 24 bits come from the R15 register and the lowest two bits are always zero.

Recommended reading for the assembly language course is Mike Ginns' Archimedes Assembly Language, from Dabs Press – see page 39. **A**

Communications Software

Tim Saxton

Triumph Over Adversity? — A review of ArcTerm and U-Connect communication software.

The Archimedes computer boasts a socket on the back panel labelled 'RS423', and 'comms enthusiasts looked forwards to the arrival of software to control it to allow them to boost BT's profits to even higher levels. Unfortunately a combination of hardware (chip level) and software bugs has made this a far from trivial task. It is only now, almost a year after the first machines were released, that reliable comms are becoming possible with the Archimedes.

This review looks at preliminary releases of the first two communications packages available, U-Connect 1.02 and ArcTerm 6.01. For reliable duplex operation of the Archimedes port, an Acorn written fix has to be installed. Without it there are a lot of errors on received data, but interestingly most of the file transfer systems still work, but v-e-r-y slowly as lots of blocks get corrupted and are rejected. Both manuals warn that the fix is not 100% effective, but I saw nothing that couldn't be put down to line noise when accessing databases, and direct links between computers never gave problems with the fix installed.

While both systems allow good comms with the Archimedes, they achieve their objectives in quite different ways, and to a rather different level. U-Connect is a quick-to-learn system without too many bells and whistles, working in the familiar Archimedes WIMP environment. All functions are selected with the mouse, although some are duplicated using <alt> key combinations. Those of you using the desktop regularly will immediately feel quite at home with the operation of U-Connect. ArcTerm on the other hand, does not use the WIMP at all, control being with <alt> key combinations, in a similar fashion to ProComm on the IBM PC. Windows do 'pop up' giving options for selection by key presses, but these are not running under the WIMP environment. This method of control takes a little longer to get used to, but there are many extra features that the real comms enthusiast will appreciate.

Both pieces of software were in 'pre-release' or very preliminary format, so there were quite a few rough edges to them, and some features were not present at all. This review therefore will have to be mainly about the basic operation, with some of the extra features not seen yet or not working as expected.

What are the basic requirements for a comms package? Firstly it should be able to control a modem. Secondly it should be able to emulate some sort of terminal and thirdly it should be able to transfer files. With these three basic functions you are in business. Both the packages under review more or less competently perform all three. The range of facilities provided is summarised in the table opposite.

Using the Software — ArcTerm

Using the <alt> key combinations quickly became second nature and a useful feature was the automatic help screens, where if you could remember the first character of a command sequence but not the second, waiting a couple of seconds after the first key press causes a help screen with the available second letter options to appear in a window. Pressing the second letter then takes you straight to the required sub-menu. Pressing the complete <alt> sequence in rapid succession skips the help screen. If you can't remember at all what key you need, <alt>+H gives the main help screen from almost anywhere in ArcTerm.

Page up and Page down select the comprehensive file transfer options, with file name lists being selectable where appropriate. Z-Modem is particularly impressive, as it tries very hard to keep your BT bill down. The block size varies depending on the line quality, moving up and down depending on the number of errors it detects. This means that on a clean line blocks can be 1024 bytes long, dropping back to 512, 256 or 128 if things get rough, but always seeking the highest suitable speed. An even nicer feature of Z-Modem is that if the line is so bad that the transfer has to be aborted, you don't lose the amount of file already safely received. Even days later you can try again and ArcTerm will say that the file already exists, and will offer various options, one of which is Continue. Selecting this simply starts where the transfer went wrong before! As files for the Archimedes tend to be rather long, even when compressed, Z-Modem is a real money saver.

The Mode 7 protocols on ArcTerm do not use Arthur's inbuilt characters at all, but instead use a much smoother set, which integrate in with the rest of the ArcTerm help and option windows. The off line editor (which is mouse driven) is a joy to use, with facilities for storing template frames with basic drawings, etc to be customised as needed. Lots of mail and information provider extras give a very good Prestel terminal facility. A 'notepad'

Basic Facilities	ArcTerm	U-Connect
Terminals emulated	Viewdata (Prestel) Teletype ANSI colour BBC mode 7 Colour VT52 most of VT100 * [†] HEX (basic data & ASCII interception)	Viewdata (Prestel) [†] Teletype VT52 most of VT100 * [†]
Modem drivers	25 drivers from dumb to full Hayes.	9 drivers from dumb to full Hayes
DIY drivers	Easy to implement	Easy to implement
File transfer protocols	X-Modem Z-Modem Kermit (multi-file) CET (Prestel type file transfer)	X-Modem Y-Modem Kermit (single file) CET

Extra Facilities	ArcTerm	U-Connect
Auto logons?	Yes, up to 3 keyword/answer sequences [#]	Yes, but only 2 [SID needs 3]
Time on-line display	Yes	Yes
Off-line text editor?	Yes [#]	Yes [*]
Command language?	Yes [#]	No
Spool screen?	Yes	Yes
Review screen?	No	Yes
Parallel printer	Yes	Yes
Scratchpad?	Yes [*]	No
Acorn RS423 patch?	Supplied	Supplied
Dynamic display of BT Line state?	Yes	Yes
Telephone Directory?	Yes	Yes
Direct dial from keyboard?	Yes	No [†]
* commands?	Yes	Yes
Cost accounting	Yes	No
Encryption of phone directory + passwords	Yes	No
Encryption of data files for transfer	Yes	No
'Host Mode' facility	Yes	No

* – not in version reviewed

– bugged in version reviewed

† – see text

allows transient data to be recorded for later retrieval while on line. (Promised but not yet implemented.)

Professional users will probably want to use the VT100 emulator with DEC mainframes, so the accuracy of the emulator can be important if trying to use WP's or programs that position the cursor using escape sequences. The smooth scroll is not implemented at all (Why not? – the smooth scroll on GraphicWriter is superb, so it can be done) and 132 column is not supported. Also in the version tested, not all escape sequences were correctly interpreted.

The command language is a BASIC-like language working from a 'script' that allows all the operations of ArcTerm to be sequenced, with conditional statements and a window function for prompts and messages. I was very quickly able to make my copy of ArcTerm autoboot on a 'script' file that presented a window with various database options to select with a keypress and then dial the selected number and perform the whole log-on sequence automatically. A supplied text editor allows the script files to be assembled, although Twin or even *build will work. There is a 'Host Mode' built in, which allows you to use your computer as a mini bulletin board – useful if you are away with a portable computer and want to access your system for files, etc. Using a combination of the inbuilt command language and the Host mode, it should be possible to write a 'script' that would automatically download the news or messages from Prestel, etc, into files on your system, and then allow you to load them remotely into your portable when you wanted, via the Host mode. Password access to either of two levels is provided with the host mode.

Using the Software – U-Connect

Good use is made of the Window and Mouse system in-built to Arthur. The main Icon select bar at the bottom allows one to click on a single help file, which explains the main features of the system, or a disc menu, allowing filing system operations. Following this is a setup option, which allows definition of Terminals, Modems, Phonebooks, etc. to be selected, modified and then saved for subsequent use. The next icon selects the status display, with a clear dynamic display at the bottom of the screen showing the current status of the Comms interface, e.g. online, speed, time connected, etc.

An onscreen keypad may be selected to emulate the 12 button Prestel pad, and the Frame/page editor can also be called up. (The latter was not implemented in

the version under review.) You can very quickly adjust the setup to suit your requirements, although dialing does need the number to be entered in a phone book first, there being no direct number entry.

The mode 7 emulation again does not use the Acorn mode 7, and is a little odd, with the standard 80 character screen being narrowed to a 40 wide window, which gives a rather odd aspect ratio, being higher than it is wide. The colours also are not standard, the shades being subtly altered as they are used in the window frames. That said, it is all perfectly legible, and with one or two minor restrictions in the version tested, seems to be a good emulation.

Again, the VT100 emulation has no smooth scroll or 132 column mode and it did not handle all the cursor positioning escape code sequences correctly in the version tested. A very nice feature is the ability to review, on line, the communications that have scrolled off the top of the screen. How often have you wanted to recall that phone number or filename that just vanished? No problem, up to 40 previous screens can be viewed in the review window, while still on line. The manual does warn of response time problems under certain conditions, such as a complicated emulation and a lot of WIMP activity, but I was not aware of any problems.

It was possible to 'hang' both ArcTerm and U-Connect by sending data to the port with no modem connected. This filled the transmission buffer and then locked up the interrupts and hence the programs, until the buffer begins to empty.

Manuals

Comms can be a minefield for even experienced users and connecting a new modem to a new computer using new software is almost bound to fail first time! Both packages realise this and try to produce a helpful manual. The manuals reviewed, like the software, were preliminary, but a real effort is made by both to help the innocent newcomer to the communications scene to succeed without too many tears. Great care is taken by both to explain as clearly as possible the tricky area of the cable between the computer and the modem. Carefully making up a cable for an 'intelligent' modem or for a 'dumb' modem as per the instructions gave success first time with both U-Connect and ArcTerm. [It is worth noting that the cables for the intelligent modem were not identical for the two software packages.]

The U-Connect manual assumes a certain knowledge of Arthur, the ADFS structure and the WIMP terminology, but explains the operation of the program simply and clearly with enough technical detail of the comms process to solve most problems. It is at the right level of

detail for a general purpose communications package. The ArcTerm manual is really very comprehensive, and the contents and appendices contain much useful comms information, with amongst other things, details of the various file transfer protocols, a section on selecting a modem and even some useful bulletin board numbers.

Conclusions

These are the first two comms packages produced for the Archimedes, so there is no direct competition to compare them with. However, the IBM PC standard comms software, ProComm, will run under the PC emulator. This has been around for a number of years, and is a mature piece of software, available for \$25-\$50 as Shareware. How does ArcTerm stand up to ProComm? Potentially, quite well. I say potentially, as the slickness of presentation and range of options on ProComm is very high compared to the current pre-release of ArcTerm, but each new version of ArcTerm I see, is slicker and more comprehensive than the last. The lack of Viewdata mode is of course a significant omission from ProComm.

U-Connect on the other hand does not set out to do everything, but elects for a more limited uncomplicated approach. Having decided to use the WIMP environment for ease of user access, it has to accept the few operational restrictions this imposes. I suspect that the main users at which this package is aimed will mainly be using Prestel type databases and will be put off by the odd screen shape and colours.

So, you pay your money and take your choice, U-Connect for an easy to use, straightforward WIMP driven system with a restricted range of facilities and a slightly eccentric mode 7 emulation, or ArcTerm for a more complicated but very comprehensive system with good Viewdata emulation. If all you want is basic comms, then either in their present state will probably do but I would like to see more of the bugs removed and missing bits provided before giving them an unreserved recommendation.

By next month we hope to have Beebug's Comms Package and I will review it and try to compare it with these two packages. A

fair enough—but if you tell it to continue it then produces a disc full error, which does not make the cause of the problem obvious. If the "bak" directory does not exist then the correct message is provided informing you that First Word Plus cannot write to the disc, and even suggests the disc may be write protected. This, until realised can tempt you not to use the auto-backup facility as it appears to be more trouble than its worth! A

First Word Plus Forum

Reaction to First Word Plus has been generally good, but there have been a number of grumbles. We were able to answer some of them ourselves, but we sent them to Acorn who were extremely helpful and gave us lots of answers. Here are various points, some with answers, some without and some asking for help from other First Word Plus users. The tone of the article is rather negative, but hopefully it is for a constructive purpose, to see if folk can find solutions to some of the problems. If you have any solutions, let us know. Thanks.

- **Palette problems** – There seems to be real confusion caused by different palettes in screens transferred from the BBC and in Artisan and in First Word Plus. Can anyone explain this properly?

- **Paging problems** – The most frequent complaint was that all attempts to make First Word Plus stop at the end of a page (for using single sheets of paper) failed miserably. This is a bug which has since been fixed. If anyone wants to use single sheets, send your disc to Customer Services at Fulbourn Road and they will send a replacement disc free of charge.

- **There is no SWAP CASE.** This is an amazing oversight, also pointed out by one or two other correspondents. (Mind you, although Wordwise on the BBC B (c.1981) had this facility, Word 3.01 (costing £200+!) that I use on the Mac hasn't!) Can anyone see a way of 'bolting on' such a facility?

- **There is no access to the *-commands.** True, but you don't need *-commands for cataloguing, mounting new discs or changing drives. If you select Open File and click on the close icon (an 'X') on the file list, it will go up one level in the directory structure – any directories in the list which it then produces have an arrow on them so, to go into one of those directories, click on the arrow.

If you want to change discs, remove the first one and click on the close icon then insert the new disc and click on the close icon again. If you just change the disk without clicking on the close icon, it will display the catalogue it has in memory, but removing the disk makes it recognise that the disk has changed, so it mounts it and re-catalogues it. If you want to change drives, click on the box containing the directory name, delete the existing name and type :1.* or whichever drive you want, press <return> and it will catalogue the root directory.

If you actually want to make it default to, say, drive 1 for the documents whilst keeping all the resources on the main disc in drive 0, you can edit the BASIC program

Library.1stWord+, (LOAD "L*.1*+") changing line 30 to something like *SET FirstWordPlus\$Docs :1.\$, and then create a "docs" directory on the second disc to contain the documents.

- **Tabs are hard, not soft.** i.e. once they are typed, they cannot be altered by changing the ruler. They must be laboriously erased and re-typed by hand. If you insert some new text on a line before a tab, the formatting is corrupted and you have to re-do the tabs!

- **No re-formatting after search and replace.** The onus is on the user to reformat after a search and replace if it changes the lengths of words.

- **Replacing "?" with ""** causes the software to crash, requiring a <ctrl-break>. It's a bug and is being checked.

- **Replace fails** if a line of more than 160 characters would be created by making the change. This would cause real problems if you were trying to work on a 160 column document.

- **Header and footer text length is limited,** it is not shown on the edit screen and you have no direct control over its style.

- **Header and footer text cannot be changed** within a file, so you need to keep, say chapter 1 and chapter 2 etc in separate files.

- **But then you cannot continuously print consecutive files.** (Even Wordwise does that – carrying the formatting, including page numbering, on from one file to the next.)

- **There are no user-definable keys** which many of us non-typists find such a boon on WPs like Wordwise.

- **Awkward handling of highlights.** If you have a paragraph with some words italicised, superscript etc and you make the paragraph bold, the other highlights are lost.

- **Character pitch cannot be changed** within a line – unless, say Acorn, you define, say, light text as a different size within a printer driver.

- **Delete and copy don't work** across a line end if the ruler is wider than the screen – yes, it's a bug and Acorn are trying to fix it.

- **Problems with auto-backup.** – If you try to save a file on write protected disc which contains the "bak" directory necessary for automatic backup then First Word Plus informs you it is unable to make a backup –

Continued opposite

Presentation Graphics

Gerald Fitton

Lingenuity claim that Presenter provides "Affordable Presentation Graphics which are compatible with Artisan, First Word Plus and GraphicWriter." This is true. It is priced at £29.84 (inclusive of VAT and UK postage - £23 through Archive) and many will find that this package is entirely adequate for their needs. However, there are some who want, and are willing to pay for, a more sophisticated package.

What can you do with it?

With Presenter you can produce a single pie chart from any one of nine sets of data, or you can produce multiple bar and line charts with up to 9 sets of data on one chart. A set of data comprises up to 25 values along the horizontal axis so, for example, you can have 9 lines on a graph with 25 points on each line. You have the choice of working in mode 12 or 20, both 16 colour modes.

I could only test mode 12 but this is the mode in which Artisan operates and the mode in which I satisfactorily transferred screens, as sprites, from Presenter to GraphicWriter and 1st Word Plus. In the package is a multi-tone screen dump for the EpsonFX and a colour dump for the Integrex. If you have a Plotmate (I have) and the Plotmate Archimedes module (I haven't, but, since the program is written in BASIC, I managed a 'patch') you can produce first class line drawings.

How easy is it to get started?

One of the good points about Presenter is that it follows Acorn's recommended protocols including *SETting system variables for the directories and the full, even perhaps over-use of the WIMP environment. If you are not familiar with these (e.g. click on the 'Adjust' button to delete the previous operation or click on the window close icon, an 'x', to close a window), then you can sometimes get stuck in a dead end.

There is a single tutorial which takes you through the main features of the package. I got stuck twice by making a simple mistake and had to start again because I couldn't find how to correct it until I'd read the whole manual. One mistake was, when I came to save a set of amended data for the second time, I clicked on the new file name only to find the software would only let me 'Load' once I had clicked on a filename. In fact, clicking <adjust> would have let me 'Save' but I didn't realise that until I'd read the fine print.

Presenter in use - Entering data

It took me several hours of practice, spread over several

days to reach a reasonable level of proficiency, but having done what I thought was reasonable, I freshened up with a cup of tea, started the stopwatch and began in earnest to produce a bar chart showing the average of 30 years weather data for England and Wales.

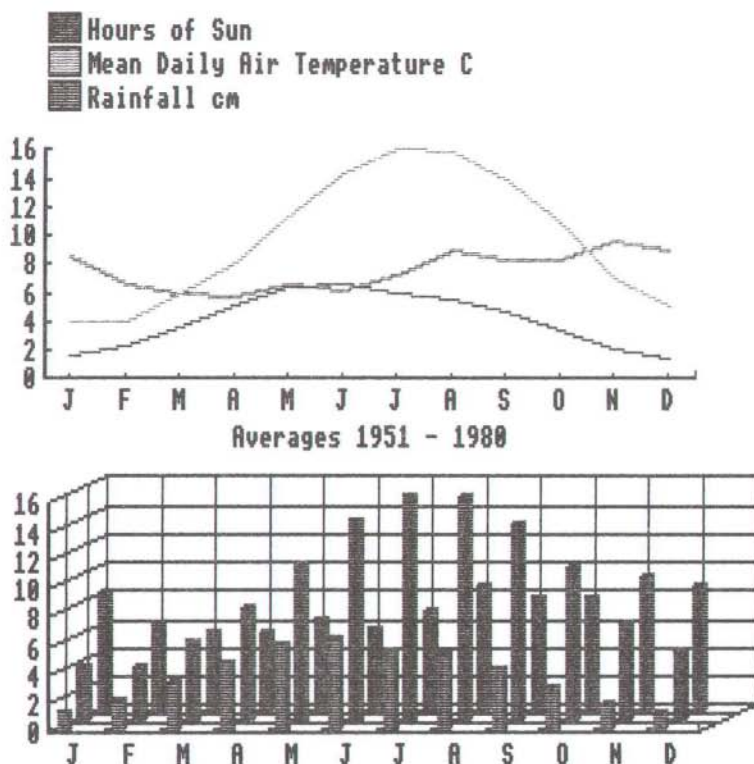
You are presented with a spreadsheet having rows numbered 1 to 25 and columns A to I. To name the file you have to Save it and then re-Load it. I did that first before putting in any data. Next I set the cell movement option ON (the default is OFF) so that when I entered some data and pressed <return> the cell moved down one row. Then, with the mouse (the cursors do nothing) I clicked on the cell at the top of the first column, headed Labels (the second column is the 'A' column) and then clicked again in a special window marked 'Value'. I typed in 'J' for January, this appeared in the Value window but, surprisingly NOT in the cell window until I pressed <return>. The highlighted cell moved down one as expected allowing me to continue with 'F', 'M', etc. There is no way of seeing what you are typing into rows 19 to 25 because the 'Value' window has disappeared off the top of the screen by then. Having typed in the monthly labels, I had to use the mouse to select cell A1 before typing in the 12 monthly values for the Hours of Sunshine, followed by the Mean Air Temperatures and the Rainfall in column C.

Configuring the Printer Options

I decided to set up the printer before moving into the chart drawing window. From experience I knew I wanted the EpsonFX, with Format Landscape (default is Portrait) and Style with both Centre (default is hard against the left margin) and Border (Without Border seems to have a 'bug' that produces a partial border). To do this requires clicking <menu> and then going down through the sub-menus 3 times. This is because selecting, for example, 'Border' clears you back to the Presenter window from which you must click <menu> and start going down through the sub-menus again.

Saving the Data

Having configured the printer options and got all the data in, it is now a good time to save the data, just in case of accidents. I clicked on 'Catalogue' and a window opened showing various data files I had created including my original blank file called Weather. From experience I knew that if I clicked on this I was in danger of loading my blank over the top of my data. No! The thing to do now is click <menu> whilst the pointer is WITHIN the catalogue window but NOT pointing at a



file name. This pops up the File Menu which has the Save option highlighted. This leads to a Save Data dialogue box with the current file name. Clicking on the dialogue box pops up a 'Confirm' dialogue window with a Yes/No option. Clicking on 'Yes' saves the file. You now need to close the Catalogue window.

The Bar Chart

I clicked the Menu button and moved along the Graph type option to pop-up the Graph menu. I clicked on Bar chart to open the Bar chart window. Clicking on the full-size box (top right of the window) fills the screen with the Bar chart window. The menu of Bar options leads to a range of sub-menus. I shall describe only some of them. So far as layout is concerned, the important one is the Margin sub-menu. Getting the left, right, top and bottom margins to my liking took a little time. I would have preferred to be able to use the mouse to define a box that could then be scaled and moved around the screen. Another sub-menu allows the colour coded key to the bars to be placed either horizontally or vertically

anywhere on the screen. Quite often, but not with this particular chart, it is necessary to type hard space into the Presenter data windows to obtain the correctly centred effect. That takes time and practice and I would have preferred to adjust the layout on the chart rather than by manipulating text from the main Presenter window. The width of the bars can be adjusted as can their colour. You have the option of 3-D bars with or without a grid.

Printing

Up to now it had taken me 27 minutes to get the chart as I wanted it. I selected Print from the Bar options menu. The dump is of good quality and a copy is shown above. The dump is written in BASIC and it took over ten minutes to dump the screen. I also set up a line graph with the same data as shown.

Saving Screens

Selecting the Save Screen option from the Bar options menu leads to a dialogue box which allows the screen to be saved as a mode 12 sprite. If you squash the window using the Adjust window size icon (bottom right) then

only the part of the chart shown in the window will be saved. Loading these mode 12 sprites into Graphic-Writer, First Word Plus or Artisan is no problem. One of the problems which does arise is that when you arrive in Artisan, you will find that the palette is different from and that of Presenter. This is particularly noticeable because Presenter's white background maps to Artisan's black. The result of a straight dump in Artisan of the same sprite is a lot of black, however, using Artisan's flood fill and magic brush it is possible to clean up the picture. You can in fact use any of the facilities of Artisan to enhance the presentation before dumping it.

Presenter's Good Points

Presenter uses Acorn protocols and makes full use of the WIMPS environment. You can enter multiple line and bar charts with up to 9 lines, each having 25 points. All or part of a screen can be saved as a sprite and loaded into other Archimedes programs. The screen dumps and the charts plotted with Plotmate are of good quality. Data can be entered in Comma Separated Value Format, e.g. from PipeDream, though for such a small spreadsheet I would be quite happy to type the data in, putting up with the 'Value' box feature. It is written in BASIC so experienced users can improve it (e.g. my Plotmate 'patch').

Presenter's Bad Points

The error trapping could be improved. It remembered my mistakes for too long! For example, in the Pie chart window I wanted to draw the pie for column 'B' ('A' is the default) and I typed 'b' by mistake. The only way I

could then draw ANY pie chart was to Save the data, restart Presenter, in fact to start again. The borderless dump has a border. The spreadsheet is limited to 25 rows so it is no good me trying to load the monthly unemployment figures for the last 30 years (360 data points) and the spreadsheet data entry has to be done 'blind' for the higher valued rows. The cursor keys can not be used as an alternative when entering data into the spreadsheet. You do have to be familiar with Acorn's WIMP environment to correct some mistakes. Finally, you can not create a screen with more than one chart on it.

Conclusion

The facilities provided by Presenter may well be sufficient for many users and at just under £30 is certainly affordable. The input is numbers into a simple spreadsheet and the charts produced are a pie from one column or multiple line or bars using all the columns. Alternatively, but for line graphs only, you can use the numbers in alternate columns as X and Y co-ordinates of (up to) 5 lines instead of plotting all 9 columns against the labels. The output is either a slow screen dump, a very acceptable plotted chart or a mode 12 sprite. The facilities for positioning text on the chart are rudimentary but just about adequate. This may be enough for many. However, there will be others who, in spite of having just bought an expensive and powerful machine, will still have a bit of cash left over and would be willing to pay more to get a package that has more features and leaves less to the ingenuity of the user. **A**

Archimedes Programmers' Utilities

Brian Cowan

This two disc set of utilities is a welcome addition to the growing body of software available for the Archimedes. One of the discs contains the MicroEMACS text editor and a collection of useful utilities and the other disc contains Kermit, the file transfer program. Three booklets come with the package: a general manual covering the utilities, a manual on EMACS and a manual on Kermit. The general manual almost matches the Guardian in the frequency of misprints! Although adequate, the manuals are certainly not over-helpful. Presumably Archimedes users are regarded as being a "serious" bunch. The general manual tells us that the Archimedes versions of Kermit and MicroEMACS together with the other utilities were written for use in the development of a "major software project for the Acorn Archimedes". One waits in excited anticipation.

Kermit

Acorn have already released their attempt at Kermit but it does not seem to work—at least not properly. For those not familiar with Kermit, this is a public domain generic program involving standardised protocols for data transfer between machines. Most computers and micros have an implementation of Kermit; this one for the Archimedes is most welcome. There are instructions for connecting an RS423 cable between the Archimedes and other systems, and users are reminded that for data transfer to actually work, the Serial Port "patch" module must be installed. (The patch was put on Archive program discs 6 and 7.)

The Kermit manual is in fact the manual of the generic version with various Archimedes additions; for example Arthur can be called from within Kermit. There is a comment that a few Kermit instructions can not be implemented; they are simply ignored, the system will

not "hang up". There is also the disclaimer that although the dial command has been included, it has not been tested! The program works a treat for transferring files between Arc and BBC but so far I have not had any success in communicating with a VAX computer.

MicroEMACS

There are various text editors available for the Archimedes including Acorn's own Twin. The Ralph Allen company chose EMACS because it is used on a wide range of machines; it has become somewhat of an industry standard. This makes it convenient for programmers to move from one installation to another. A summary is given in the general manual: "Emacs is a full screen text editor. It has the ability to edit several files at once placed into multiple windows on the screen, together with powerful search and replace commands and magic mode for auto-indentation for C programs if required." The EMACS manual is the standard EMACS manual and it is quite comprehensive. It also has additions relating to specifically Archimedes additions. I suspect that as there are quite a few text editors around for the Archimedes, EMACS will be of greatest use to

those who have used to the Archimedes from other systems where EMACS was used.

Other Utilities

Most of the utilities relate to the manipulation of files. There are file comparison and search functions and a means of reading just the start or the end of a file. A useful feature is a compaction command, called CONTIG. Not a lot of people know this, but the *COMPACT of the ADFS does not necessarily fully compact a disc; you may need to do it a number of times. However, CONTIG takes care of this in one go. Although these utilities are most useful, they are all written as absolute code. They can therefore not reside in RAM in a relocatable module, but must be loaded in from disc as and when called. From a floppy disc this is slow. It is faster from a Winchester and a RAM disc would be faster. RAM module owners can of course use the RAM Filing System. Obviously this collection of utilities would benefit from implementation as a relocatable module.

Programmers Utilities costs £38.80 inclusive from Ralph Allen Engineering. **A**

ARCHIMEDES SOFTWARE TOOLS

Programmers utility disk N°1 includes:

MicroEMACS: The UNIX industry standard text editor.

Supports Multi-file working, Multi-window, user macros. On-line help facilities.

Access to the operating system.

Screen resolution changeable whilst editing.

KERMIT: The industry standard intelligent file transfer utility.

Block checking with automatic re-send of faulty blocks

(end those RS423 problems!). Multiple file batch transfers,

access to the operating system. On-line help facilities.

Plus

UNIX type: HEAD, TAIL, TR, GREP, UUENCODE, UUDECODE and more...

All this for only £38.80 including discs, manuals, VAT and postage

ACCESS / VISA by phone or send cheque to:-

Ralph Allen Engineering Co., Forncett-End, Norwich,

Norfolk, NR16 1HT. Tel. (Bunwell) 095389-420

Logotron Logo for Archimedes

Matthew Treagus

For many years now the trend in British schools has been away from teaching the traditional languages such as the "undisciplined" BASIC and the "too much like hard work" machine codes. The emphasis is moving towards the more structured, precision languages in order to teach techniques instead of the actual language. Amongst these "new educational languages" are Logo, various control languages, PASCAL and at higher academic levels, ARM Code, C and 68000 assembler.

Why Logo?

All of the above educational languages have advantages and disadvantages but Logo seems to have a more balanced array of attributes. Logo is the very much an "all-rounder" and caters "for all ability ranges" from 5-18 year olds. At one end of the scale, Logo can simply draw on the screen and introduce the principles of angles and begin to develop computer awareness. At the other end of the scale "mainframe" Logo (with the help of some extensions) is capable of using multi-turtles (a hint multi-tasking!), of list handling (a dash of LISP) and much more. Installing primitives allows Logo to be extended to meet the needs of the user. In addition, Logo has hooks on which real-world applications can be hung such as 3D modeling, sound, music, control applications and development packages.

Logotron Logo for the Archimedes

Logotron have always been leaders in the field of Logo on Acorn machines and indeed they are the first to release Logo for the Archimedes. The advertising for it gives a very impressive and accurate over-view of the entire package. Logotron say,

"Mainframe 32 bit Logo Performance - on a micro". This, for a start, injects a higher degree of usability giving an educational language that runs at a reasonable speed. (This must be a first on any micro!!) Logotron are claiming speeds 10 times faster than the BBC (not really surprising but reasonably impressive!)

"Largest ever Logo workspace on a micro". Again, this is not really surprising but just an advantage of the Archimedes over other micros. Never-the-less the workspace can be up to 4 Mbyte now and could possibly be 16 Mbyte (RAM Chips permitting).

"Full support of the Acorn Floating Point Unit". Nice to see a bit of upwards compatibility. The disc is supplied with the latest version of the Floating Point Emulator which is used if a Floating Point Unit is not installed.

What do you get?

Logotron's Archimedes Logo is supplied on a single disk with a Reference Manual and a Tutorial Manual. The manuals were of an exceptionally high quality and covered the entire spectrum of ability from the unwilling to the whizz-kid. The Reference Manual was outstandingly impressive, some 280 pages devoted to a big and bold Logo dictionary of the 200+ commands. A registration form is enclosed that entitles you to a spare backup copy of the program disc. The program disc contains several demo Logo programs. Two extensions are also included to allow the use of multi-turtles and interaction with the mouse. A set of additional books containing various Logo resources and projects are available at £6 each or £20 for the set of 4.

Does Archimedes Logo meet the standards?

In a word YES! It does meet the standards perfectly and it is compatible with BBC Logo (more or less). The principles of Logo are well implemented - the idea of flexibility is enforced by the ability to "bury" procedures into Logo's workspace, effectively creating more primitives. This means that Logo can be tailored to suit the individual needs of the establishment. As for speed, I have already said that Archimedes Logo is the fastest around. The PCW Benchmarks speak for themselves...

BBC B Logotron Logo	- 49.75 secs
IBM Logo	- 21.5 secs
Archimedes Logo	- 5 secs

The Future

Logo has always been widely thought of as the best of the educational languages - quite an honour for any package. However, in the past Logo has often been frustratingly slow but now, with the power of the Archimedes, its speed has been brought up to a level of acceptability. I am sure that we can expect many more developments in the area of Logo over the next few years as Logotron continue to work on this evolving package. An extensions editor will be available later in the year although I would imagine supply would be limited to recognised Logo approved companies to ensure a continuing high standard of Logo related products.

Conclusions

I was very impressed not just with the Archimedes Logo program but also with the entire package from documentation to the user support offered. This is an unrivalled package and every school should have one. It also comes recommended for any home user with

GraphicWriter/Wordwise Conversions

Glyn Emery

Wordwise Plus is all right, particularly if you have been using it for a time; but there are some things it does rather slowly under the emulator. GraphicWriter offers a few more facilities so you may find it worth while to convert some of your files from the one to the other. Here is my technique for doing this:

1. Spool the Wordwise text. I find it best to do this on the original disc; but I create a new directory for the spooled files.
2. Wordwise files are data type (FFD). Graphic Writer will only accept its own and text type files (FFF). First of all, therefore, you must convert the spooled file with: *Ssettype spooledfile FFF
3. Change discs and load Graphic Writer.
4. Put back the disc containing the spooled file.
5. Use the mouse menu key to get the second main menu. Click <select> on the file icon, and enlarge the file window if necessary. Click <menu> with the arrow in the file window to get the file menu. Clicking <select> on the blue file icon should mount the disc containing the spooled file. The file window should now be showing the root directory of this disc.
6. If the spooled file is in a sub-directory you will have to click on the name of this to get the filename. Click on the name of the spooled file to load it. Loading is very slow indeed. Presumably text files have to be processed quite a lot before Graphic Writer will accept them. Go off and get yourself a cup of coffee.
7. You can now edit the spooled file. At the very least you will want to add a ruler at the start. You can also preview the file if you want to.
8. Put back the GraphicWriter disc, or the disc you want to keep your GraphicWriter files on, and repeat the operations in step 5 to mount it.
9. Save the file by clicking on TEXT in the file window and typing in the new filename. You may get the message that the file already exists; but ignore this.
10. To check that the file was saved OK, reload it.

children who wish to explore the exciting world of computer programming.

The package costs £60. A site licence is available at £300 and LEA licences available on request. These prices exclude p & p and VAT.

Thanks to:- Hardley School Computer Department for their help with this article. M.T. **A**

A caveat...

The working space available for GraphicWriter may be less than that available for Wordwise Plus, particularly if you have reserved some space for sprites. You will get no warning if the spooled file is too long. GraphicWriter will just load as far as it can; so you will have to go back to Wordwise Plus, split the file, and repeat the exercise. You can find how much space you have by clearing the GraphicWriter text (X icon) and clicking on the 'pen' icon, when a window full of vital statistics will appear.

Converting back again

Conversion the other way, from GraphicWriter to Wordwise Plus is rather easier. If you click on the TEXT icon in the file window with <adjust> instead of <select> you can save any GraphicWriter file as a text file. Type in the name for the new file and click on OK, when the file will be saved. Its name then appears in the text list in the window in a "Cambridge" instead of an "Oxford" blue strip. A file stored in this way can be read directly into Wordwise Plus. It may have a few "pad" characters scattered around; but these can be removed simply with a search and replace command. **A**

continued from page 27

```
1400 WHEN 3 : PROCclose_window
                                (!block%)
1410 WHEN 6 : IF (block%!8)AND2>0
                                THEN PROCmain_menu : REM if
                                MENU pressed pop up the menu
1420 WHEN 9 : PROCmenu_action :
                                REM decode the menu

1430 ENDCASE
1440 ENDPROC
1450
1460 DEFFNwimp_initialise
1470 LOCAL version%
1480 SYS "Wimp_Initialise",0 TO
                                version%

1490 =version%/100
1500
1510 DEFPROCOpen_window(handle)
1520 block%!0=handle
1530 SYS "Wimp_OpenWindow",,block%
1540 ENDPROC
1550
1560 DEFPROCclose_window(handle)
1570 !block%=handle
1580 SYS "Wimp_CloseWindow",,block%
1590 ENDPROC
```

continued overleaf

More Comments about SigmaSheet

Rob Brown

Rob has been continuing to use SigmaSheet since he wrote his review and has come up with some more comments which should hopefully prove useful both to prospective purchasers and to existing users.

- As mentioned in my review, I found the documentation describing the printing function (pages 52-54 in the manual) to be somewhat confusing; in particular the examples shown in the middle of page 53 show single quotation marks being used whereas I found that double quotes had to be used or alternatively the ASCII code equivalent.

- As also mentioned in my review I found various problems with the DeltaLink import function, in particular:

1. The sub-menu shown at the top of page 56 of the manual does not match the screen display.
2. Contrary to page 56 of the manual, only 5 lines appear on the screen display within the Define Fields table; I was advised by Minerva that the numeric value within line 580 of the Basic program "SS-Totals" should be amended from 5 to 16 to change this (and it did!).
3. The Define Fields note on page 57 indicates that a 'value of' field (eg. #sal) should be classified as type T (i.e. Text). I found that it had to be classified as type S (i.e. Sum).
4. Option W to Wipe Definition shown on page 58 does not appear on the screen display nor does pressing the W key work either!

5. To achieve a straightforward output of the whole file within the Define Transfer, I was advised by Minerva that the word NIL should be entered at the Grouped By? prompt, this is not documented at all on pages 58 or 59.

- A further minor annoyance, which has now become apparent since my review, concerns the various sub-menus; there is no easy way to return to the preceding menu because pressing <escape> simply returns you to the main spreadsheet.

- As noted in the June issue of Archive, other readers have experienced some further problems which I had not encountered at the time of my review. If I may offer some help on the second problem to do with "holding" a line on the screen. I find that the facility to rename Column and/or Row titles (documented on page 23 of the manual) goes some way towards the Inter-Sheet "hold" facility and, in my view, the printing feature of Fixed Rows/Columns within SigmaSheet is somewhat easier to use than the equivalent in Inter-Sheet.

- On a value for money comparison, the list price for SigmaSheet is £69.95 and that for Inter-Sheet is £44.85; but to the latter may well have to be added the cost of both the Backplane and the ROM/RAM Podule. I wonder to what extent Minerva are copying the car manufacturers in deliberately leaving some features out of the "middle of the range" SigmaSheet which will be incorporated within the "top of the range" System Sigma, as and when that product becomes available! **A**

```
1600
1610 REM decode the item_selected_list
1620 DEFPROCmenu_action
1630 level0=block%10
1640 level1=block%4
1650 CASE level0 OF
1660 WHEN 0 :first=NOT first
1670 WHEN 1 :second=NOT second
1680 WHEN 3 :CASE level1 OF
1690     WHEN 0,1 :sel=NOT sel
1700     WHEN 2:PROCclose_down
1710     ENDCASE
1720 ENDCASE
```

```
1730 ENDPROC
1740
1750 DEFPROCmenu_close
1760 SYS "Wimp_CreateMenu",,-1
1770 ENDPROC
1780
1790 DEFPROCclose_down
1800 SYS "Wimp_CloseDown"
1810 MODE 12
1820 END A
```


Fact-File

ABACUS Training	29 Okus Grove, Upper Stratton, Swindon, Wilts, SN2 6QA.
ACE Computing	27 Victoria Road, Cambridge, CB4 3BW. (0223-322559)
Alpine Software	P.O.Box 25, Portadown, Craigavon, BT63 5UT. (0762-42510)
Atomwide	(0689-38852)
Blue Grey Software	28 Bobbitts Way, Wivenhoe, Essex, CO7 9NJ.
Cambridge Microsystems	19 Panton Street, Cambridge, CB2 1HL. (0223-66553)
CCD Computer Services	71 Marlborough Park Avenue, Sidcup, Kent, DA15 9DL. (01-302-5427)
Clares Micro Supplies	98 Middlewich Road, Rudheath, Northwich, Cheshire, CW9 7DA. (0606-48511)
Colton Software	149-151 St Neots Road, Hardwick, Cambridge, CB3 7QJ. (0954-211472)
Computer Concepts	Gaddesden Place, Hemel Hempstead, Herts, HP2 6EX. (0442-63933)
Computerware	55 Romulus Court, Brentford Dock, Justin Close, Brentford, Middlesex.
Dabs Press	76 Gardner Road, Prestwich, Manchester, M25 7HU. (061-773-2413)
Dudley Micro Services	32 Osborne Road, Penn, Wolverhampton, WV4 4AY. (0902-342214/ 334315)
EMR Ltd	14 Mount Close, Wickford, Essex, SS11 8HG. (0702-335747)
Intelligent Interfaces	43B Wood Street, Stratford-on-Avon, CV37 6JQ. (0789-415875)
Lingenuity	P.O.Box 10, Halesworth, Suffolk, IP19 0DX. (0986-85-476)
Logotron	Dales Brewery, Gwydir Street, Cambridge, CB1 2LJ. (0223-323656)
Magenta Research Ltd	Amp House, Dingwall Road, Croydon, CR0 9XA. (01-681-7179)
Minerva Systems	69 Sidwell Street, Exeter, EX4 6PH. (0392-37756)
P.Taylor Software	23 Limes Avenue, Alfreton, Derby, DE5 7DW. (0773-834893 or 768899)
Ralph Allen Engineering Co	Forngett End, Norfolk, NR16 1HT. (095-389-420)
RESOURCE	Exeter Road, Doncaster, DN2 4PY. (0302-63800/63784)
Smith & Wiggins	77 Edward Road, Fleckney, Leicester, LE8 0AD.
Stream Software	71 Carol Crescent, Halesowen, West Midlands, B63 3RR.
Texellence	2 Greenhill Road, Coleraine, N Ireland, BT51 3JE.
Wight Scientific	44 Roan Street, London, SE10 9JT. (01-858-2699)

Norwich Computer Services 18 Mile End Road, Norwich, NR4 7QY. (0603-507057)

Archive

Subscription Magazine and Support Group for *Archimedes* users

Archive Magazine contains:

- News
- Reviews
- Hints and Tips – a major feature
- Articles for Beginners
- The Latest Technical Information
- Program Listings
- Free Small Ad's Section
- HELP – Requested and Offered
- Contact Box – to help you form common interest groups

Eureka! – Bulletin Board

0603-250689 on 1200/75 or 300/300

- Very Latest News
- Down-load Software
- Mailbox Facilities
- Chat line

Technical Help Service (£8 / year)

A telephone hot-line service for immediate help with your technical problems. Any member can send written enquiries, but for a fast response use the THS!

Members' Discount: 7.5% off software from Computer Concepts, Minerva Systems and Clares Micros Supplies purchased through Norwich Computer Services.

Subscription: 12 issues £12.50 (UK)
Europe £18, Middle East £22,
America / Africa £25, Elsewhere £27.
Technical Help Service £8

N.B. All earlier issues have now been re-printed – you may back-date your subscription as far as issue 1 (October 1987) – to take advantage of this huge bank of information.

Archimedes is a trademark of Acorn Computers Ltd.

* Please send copies of *Archive* magazine for one year starting from
Volume 1 Issue _____

* Please enrol me on the Technical Help Service for one year. (£8)

I enclose a cheque for £ _____ payable to "Norwich Computer Services".

Name: _____

Address: _____

Postcode: _____

Norwich Computer Services, 18 Mile End Road, Norwich, NR4 7QY